



DEVELOPMENT OF A LOW-LATENCY,  
HIGH DATA RATE, DIFFERENTIAL GPS  
RELATIVE POSITIONING SYSTEM FOR  
UAV FORMATION FLIGHT CONTROL

THESIS

Stephen J. Comstock, Commander, USN

AFIT/GAE/ENG/06-03

DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY

**AIR FORCE INSTITUTE OF TECHNOLOGY**

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, United States Navy, Department of Defense, or the United States Government.

DEVELOPMENT OF A LOW-LATENCY,  
HIGH DATA RATE, DIFFERENTIAL GPS  
RELATIVE POSITIONING SYSTEM FOR  
UAV FORMATION FLIGHT CONTROL

THESIS

Presented to the Faculty  
Department of Electrical and Computer Engineering  
Graduate School of Engineering and Management  
Air Force Institute of Technology  
Air University  
Air Education and Training Command  
In Partial Fulfillment of the Requirements for the  
Degree of Master of Science in Aeronautical Engineering

Stephen J. Comstock, B.S., M.S.  
Commander, USN

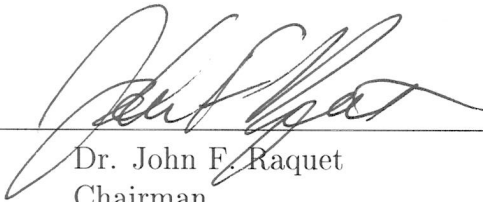
September 2006

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.


DEVELOPMENT OF A LOW-LATENCY,  
HIGH DATA RATE, DIFFERENTIAL GPS  
RELATIVE POSITIONING SYSTEM FOR  
UAV FORMATION FLIGHT CONTROL

Stephen J. Comstock, B.S., M.S.  
Commander, USN

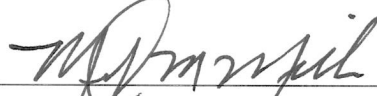
Approved:

  
\_\_\_\_\_  
Dr. John F. Raquet  
Chairman

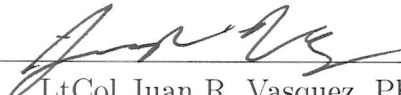
29 AUG 06  
date

  
\_\_\_\_\_  
Dr. Richard G. Cobb  
Member

28 Aug 2006  
date

  
\_\_\_\_\_  
Dr. Mikel M. Miller,  
Member

28 Aug 2006  
date

  
\_\_\_\_\_  
LtCol Juan R. Vasquez, PhD  
Member

28 Aug 2006  
date



*Abstract*

In order for Unmanned Aerial Vehicles (UAVs) to be able to fly missions currently performed by manned aircraft, they must be able to conduct in-flight refueling. Additionally, significant fuel savings can be realized if multiple UAV's are able to fly in precise formation and align wingtip vortices. In either case, the precise relative position between the aircraft must be known to an accuracy of only a few centimeters.

Previous research at the Air Force Institute of Technology culminated in the development of a relative positioning system for manned aircraft. This thesis presents the development of the next-generation system designed for small UAV's. Because of the stringent size, weight, and power consumption requirements inherent in small UAV's, several approaches were taken to maximize efficiency and performance while simultaneously keeping the system small and lightweight.

At the core of the Differential GPS (DGPS) application presented in this thesis are three separate tasks which operate asynchronously yet share information when required. A Kalman filter task operates continuously at a 1 Hertz rate. An ambiguity resolution task, utilizing the Least squares AMBiguity Decorrelation Adjustment (LAMBDA) method, is run whenever the floating point ambiguities must be resolved to their integer values. A high-rate output task, operating at a 20 Hertz rate, formulates a high-rate, centimeter-level, relative position solution with less than 10 milliseconds of latency. The use of widelane measurements generally resulted in a 2 second convergence time for ambiguity resolution and a 99.9 percent success rate of selecting the proper ambiguity set. However, in order to minimize the increased errors associated with multipath, the system quickly transitions from widelane mode to narrowlane mode. The system was tested on the ground in both a static and dynamic environment. Unfortunately there was inadequate time to conduct flight testing using radio controlled aircraft to simulate small UAV's.

## *Acknowledgements*

First and foremost, I owe a large debt of gratitude to Dr. John Raquet. Without his guidance and supervision, development of the system presented in this thesis would have been nearly impossible. He was always there to answer questions, provide insight, and keep me pointed in the right direction.

Special recognition is also due to Mr. Don Smith. As the hardware expert, he consistently helped me sort through the myriad of issues that arose which ranged from COM ports and wireless serial data links to PC-104 and GUMSTIX technology.

Last, but certainly not least, I must thank my wife and son for their continuous support. Despite all the long nights spent writing C++ code and battling with LaTeX, they never complained about me working too much or spending too much time away from them. After four deployments, the last one an eight month combat cruise, I firmly believe that one of the toughest jobs in life is that of a Navy Wife! The constant support and encouragement they provide is nothing short of incredible.

Stephen J. Comstock

## *Table of Contents*

	Page
Abstract . . . . .	iv
Acknowledgements . . . . .	v
Table of Contents . . . . .	vi
List of Figures . . . . .	ix
List of Tables . . . . .	xi
List of Symbols . . . . .	xii
List of Abbreviations . . . . .	xiii
 I. Introduction . . . . .	 1-1
1.1 Background . . . . .	1-1
1.2 Problem Definition . . . . .	1-4
1.3 Related Research . . . . .	1-5
1.4 Assumptions . . . . .	1-7
1.5 Methodology . . . . .	1-7
1.6 Thesis Overview . . . . .	1-8
 II. Background . . . . .	 2-1
2.1 Overview . . . . .	2-1
2.2 General GPS Theory . . . . .	2-1
2.2.1 Basic Principles . . . . .	2-1
2.2.2 GPS Measurements . . . . .	2-2
2.2.3 Single Differencing . . . . .	2-5
2.2.4 Double Differencing . . . . .	2-7
2.2.5 Virtual Measurements . . . . .	2-8
2.3 Carrier-Phase Ambiguity Resolution . . . . .	2-9
2.3.1 Basic Principles . . . . .	2-9
2.3.2 The LAMBDA Method . . . . .	2-10
2.4 Kalman Filtering . . . . .	2-12
2.4.1 Basic Principles . . . . .	2-13
2.4.2 Kalman Filter Equations . . . . .	2-13
2.5 Hardware Components . . . . .	2-17
2.5.1 GPS Receiver . . . . .	2-17

	Page
2.5.2 Antenna . . . . .	2-18
2.5.3 Serial Data Link . . . . .	2-19
2.5.4 CPU / Power supply . . . . .	2-20
2.6 Summary . . . . .	2-20
III. Development of Instrumentation Package . . . . .	3-1
3.1 Overview . . . . .	3-1
3.2 Top-Level Design . . . . .	3-1
3.3 Real-Time Considerations . . . . .	3-3
3.3.1 Loss of Non-Base SV . . . . .	3-5
3.3.2 Loss of Base SV . . . . .	3-6
3.3.3 Gain of SV . . . . .	3-10
3.4 Front-End Parser . . . . .	3-11
3.5 Floating Point Kalman Filter . . . . .	3-14
3.5.1 Initialization of State Vector . . . . .	3-15
3.5.2 Initialization of State Vector Covariance Matrix . . . . .	3-17
3.5.3 Initialization of Transition Matrix . . . . .	3-20
3.5.4 Initialization of Discrete Noise Matrix . . . . .	3-22
3.5.5 Initialization of Measurement Covariance Matrix . . . . .	3-24
3.5.6 Implementation of Real-Time Considerations . . . . .	3-27
3.5.7 Time Propagation of State Vector and State Vector Covariance Matrix . . . . .	3-28
3.5.8 Formulation of Measurement Equation . . . . .	3-28
3.6 Carrier-Phase Ambiguity Resolution . . . . .	3-32
3.7 High-Rate Relative Position Output . . . . .	3-34
3.7.1 Formulation of Relative Position Vector . . . . .	3-34
3.7.2 Additional Real-Time Complexity . . . . .	3-38
3.8 Summary . . . . .	3-40
IV. Results and Analysis . . . . .	4-1
4.1 Testing Methodology . . . . .	4-1
4.1.1 Kalman Filter Task . . . . .	4-1
4.1.2 Ambiguity Resolution Task . . . . .	4-1
4.1.3 high-rate Output Task . . . . .	4-2
4.2 Truth Values . . . . .	4-2
4.3 The Benefit of DGPS - An Example . . . . .	4-3
4.4 Static Test Results and Analysis . . . . .	4-5
4.4.1 Kalman Filter Performance . . . . .	4-6
4.4.2 Ambiguity Resolution Performance . . . . .	4-11
4.4.3 High-Rate Output Performance . . . . .	4-16
4.5 Dynamic Test Results and Analysis . . . . .	4-28

	Page
4.6 Asynchronous Input Analysis . . . . .	4-32
4.7 Summary . . . . .	4-33
V. Conclusions and Recommendations . . . . .	5-1
5.1 Overview . . . . .	5-1
5.2 Conclusions . . . . .	5-3
5.3 Recommendations . . . . .	5-6
Appendix A. Software Documentation . . . . .	A-1
A.1 File Listing . . . . .	A-1
A.2 Software Organization . . . . .	A-2
A.3 Implementation Aspects . . . . .	A-4
Bibliography . . . . .	1

## *List of Figures*

Figure		Page
1.1.	System Architecture . . . . .	1-4
2.1.	Single-Difference Measurement: Two receivers and one SV . . .	2-6
2.2.	Double-Difference Measurement: Two receivers and two SVs . .	2-8
2.3.	NovAtel OEM4-G2L GPS Receiver Card [11] . . . . .	2-17
2.4.	ANTCOM 3G1215X1000 40 dB GPS Antenna [10] . . . . .	2-19
2.5.	Freewave FGR09CS Wireless Modem [4] . . . . .	2-20
3.1.	Top Level Design of Instrumentation Package . . . . .	3-2
3.2.	Formulation of H matrix . . . . .	3-30
4.1.	Static Testing, ENU Errors: Absolute Position (non-DGPS solution). . . . .	4-4
4.2.	Static Testing, ENU Errors: Comparison Between Absolute Position (non-GPS) and Relative Position (DGPS). . . . .	4-5
4.3.	Static Testing, Comparison of Relative Position Error to Filter Computed Covariance Matrix (1 data run). . . . .	4-6
4.4.	Static Testing, Comparison of Relative Position Error to Expected Error Range (50 data runs). . . . .	4-7
4.5.	Static Testing, Comparison of Floating Point double-difference Carrier-Phase Ambiguity Errors to Expected Error Range (4 Separate Data Runs) . . . . .	4-9
4.6.	Static Testing, East-North Errors. Comparison of Filter Errors to Single Point Errors. . . . .	4-10
4.7.	Static Testing, ENU Errors. Comparison of Filter Errors to Single Point Errors. . . . .	4-11
4.8.	Static Testing, Comparison of Squared Normal Values. Four Data Runs (Filter Reset) . . . . .	4-12

Figure		Page
4.9.	Static Testing, Comparison of Squared Normal Values. Four Data Runs (No Filter Reset) . . . . .	4-14
4.10.	Static Testing, high-rate Output: Comparison of Horizontal Errors and Vertical Errors (Widelane Measurements) . . . . .	4-17
4.11.	Static Testing, high-rate Output: Biases Found in ENU Errors (Widelane Measurements) . . . . .	4-18
4.12.	Static Testing, High-Rate Output: Periodicity of Multipath Errors (No Offset) . . . . .	4-19
4.13.	Static Testing, High-Rate Output: Periodicity of Multipath Errors (4 Minute Offset) . . . . .	4-20
4.14.	Static Testing, High-Rate Output: Comparison of Accuracy (Widelane Measurements vs. L1 only Measurements) . . . . .	4-21
4.15.	Static Testing - System Latency . . . . .	4-25
4.16.	Static Testing - System Latency, Expanded View . . . . .	4-25
4.17.	Static Testing - Interval of Relative Position Solution . . . . .	4-27
4.18.	Dynamic Testing - Horizontal Relative Position . . . . .	4-29
4.19.	Dynamic Testing - Kalman Filter: Relative Position . . . . .	4-31
4.20.	Dynamic Testing - Kalman Filter: Relative Velocity . . . . .	4-31
4.21.	Effect of Asynchronous Data: ECEF Errors (Expanded View) .	4-33

## *List of Tables*

Table		Page
1.1.	Typical GPS Accuracy [16] . . . . .	1-3
2.1.	NovAtel OEM4-G2L Specifications [11] . . . . .	2-18
2.2.	ANTCOM 3G1215X1000 Specifications [1] . . . . .	2-19
2.3.	FreeWave FGR09CS Specifications [4] . . . . .	2-20
3.1.	Initial Covariance Values . . . . .	3-20
3.2.	Initial Noise Values . . . . .	3-25
3.3.	Initial Values for Measurement Covariance Matrix . . . . .	3-27
4.1.	Truth Values Utilized During Ground Static Testing . . . . .	4-3
4.2.	Ambiguity Resolution Performance (Widelane Measurements) .	4-15
4.3.	Ambiguity Resolution Performance (L1 Measurements) . . . . .	4-16
4.4.	Statistical Analysis of High-Rate Output (ENU Coordinates) .	4-23
4.5.	Standard Deviation for OPUS Truth Values (ENU Coordinate Frame) . . . . .	4-23
4.6.	Static Testing: Typical DRMS Values (ENU Coordinate Frame)	4-23
A.1.	Software Development: File Listing . . . . .	A-1
A.2.	Kalman Filter Thread: List of Subroutines . . . . .	A-3



## *List of Symbols*

Symbol		Page
$L1$	Primary GPS Signal (1575.42 MHz) . . . . .	2-2
$L2$	Secondary GPS Signal(1227.60 MHz) . . . . .	2-2
$\rho$	GPS pseudorange measurement (meters) . . . . .	2-3
$r$	Actual range from receiver to satellite (meters) . . . . .	2-3
$c$	Speed of light (meters / second) . . . . .	2-4
$\delta t_r$	Receiver clock error (seconds) . . . . .	2-4
$\delta t_{sv}$	Satellite clock error (seconds) . . . . .	2-4
$T$	Tropospheric error (meters) . . . . .	2-4
$I$	Ionospheric error (meters) . . . . .	2-4
$m_\rho$	Multipath error of pseudorange measurements (meters) . .	2-4
$v_\rho$	Noise error of pseudorange measurements (meters) . . . .	2-4
$\phi$	GPS carrier-phase measurement (cycles) . . . . .	2-4
$\lambda$	Carrier-phase wavelength (meters / cycle) . . . . .	2-4
$m_\phi$	Multipath error of carrier-phase measurement (meters) . .	2-5
$v_\phi$	Noise error of carrier-phase measurement (meters) . . . .	2-5
$N$	Carrier-phase integer ambiguity (cycles) . . . . .	2-5
$\phi_A^k$	Phase measurement between receiver A and satellite k . .	2-6

## *List of Abbreviations*

Abbreviation		Page
DGPS	Differential GPS . . . . .	1-1
UAV's	Unmanned Aerial Vehicles . . . . .	1-1
GPS	Global Positioning System . . . . .	1-1
IFR	In Flight Refueling . . . . .	1-1
DRMS	Distance Root Mean Squared . . . . .	1-3
IMU	Inertial Measurement Unit . . . . .	1-5
AFIT	Air Force Institute of Technology . . . . .	1-6
RTK	Real Time Kinetic . . . . .	1-6
MMAE	Multiple Model Adaptive Estimator . . . . .	1-6
SVs	Satellite Vehicles . . . . .	2-1
LAMBDA	Least-squares AMBiguity Decorrelation Adjustment . . . .	2-10
OPUS	Online Positioning User Service . . . . .	2-18
ECEF	Earth Centered Earth Fixed . . . . .	3-15
FOGMA	First Order Gauss-Markov Approximation . . . . .	3-23
OPUS	Online Positioning User Service . . . . .	4-2
ENU	East-North-Up . . . . .	4-3

# DEVELOPMENT OF A LOW-LATENCY, HIGH DATA RATE, DIFFERENTIAL GPS RELATIVE POSITIONING SYSTEM FOR UAV FORMATION FLIGHT CONTROL

## I. Introduction

This thesis describes the research effort leading to the development of an instrumentation package which combines the use of Differential GPS (DGPS), the resolution of GPS carrier-phase ambiguities, and Kalman filtering. The primary goal was to design the system in such a way as to provide real-time, low-latency, 20 Hertz precise relative position data which is required for close formation control of multiple Unmanned Aerial Vehicles (UAV's). Additionally, because the system was designed for small UAV's, emphasis was placed on minimizing the size, weight, and power consumption.

### *1.1 Background*

In the case of formation flight between multiple manned aircraft, the relative position of the wing aircraft with respect to the lead aircraft is visually determined by the pilot. This position is then compared to the desired relative position, and flight control inputs are made to correct any deviations. However, for unmanned flight, the determination of the relative position must be made via different means. Because of the accuracy of the navigation solution provided to users of the Global Positioning System (GPS), one may initially consider it as a possible solution. However, even with a dual frequency military receiver, the stand-alone accuracy of GPS is on the order of 3-5 meters [9]. In order for UAV's to be able to fulfill some of the missions typically flown by manned aircraft, they must be capable of in-flight refueling (IFR). Additionally, significant fuel savings can be realized through the drag reduction that occurs when aircraft fly in close formation and align aircraft vortices [22]. When

considering potential requirements such as these, the accuracy of 3-5 meters provided by stand-alone GPS is clearly inadequate. In fact, for multiple small UAV's flying in close formation, the required accuracy of the relative position vector is on the order of a few centimeters!

Many of the errors inherent to stand-alone GPS do not change significantly over relatively short baselines. As such, numerous techniques have been developed that utilize Differential GPS to greatly improve performance by using a reference receiver, whose position is known, thereby reducing or eliminating many of these sources of error. When using DGPS, receiver clock error and satellite clock error are completely eliminated, and satellite position error, ionospheric error, and tropospheric error are significantly reduced. It must be noted that these benefits do come at a cost. Both multipath and measurement noise errors are increased when utilizing DGPS. However, even with the slight degradation due to multipath and noise, DGPS still offers significant improvements to overall accuracy [13].

The two types of measurements that can be used by DGPS are code measurements and carrier-phase measurements. Code measurements are derived from the pseudoranges between the GPS antenna and the respective satellites. In contrast, carrier-phase measurements are obtained by keeping track of the accumulated Doppler of the GPS carrier signal. Since the wavelength of the C/A code is approximately 290 meters, compared to a wavelength of approximately 0.2 meters for the carrier-phase, the code measurements are not nearly as precise as the carrier-phase measurements. However, code measurements are known absolutely. One merely has to take the time difference between the transmit time and the receive time and multiply by the speed of light to calculate the pseudorange. In contrast, carrier-phase measurements are relative measurements. This is because the number of carrier cycles at the beginning of accumulation is not known and is frequently referred to as the carrier-phase ambiguity. In order to take full advantage of the precision of carrier-phase measurements, the phase ambiguities must be resolved to their integer values. It should be noted that both code measurements and carrier-phase measurements are used to determine

distances. However, the difference between the two can be loosely conceptualized as follows: 1) Using code measurements is analogous to using a traditional tape measure that has labelled tick marks, but a large distance between each tick (290 meters!), 2) Using carrier-phase measurements is analogous to using a tape measure whose tick marks are very close together (0.2 meters), but lack the labels.

A frequently utilized method of expressing GPS accuracy is through the use of Distance Root Mean Squared (DRMS) values which represent the two dimensional horizontal accuracy. The DRMS value is defined as the square root of the average of the square errors that exist in the horizontal plane. A summary of the expected accuracy of several modes of GPS is presented in Table 1.1 [16]. The accuracy obtained through the use of precise carrier-phase DGPS is readily apparent when compared to other modes of operation.

Table 1.1: Typical GPS Accuracy [16]

	Mode	Horizontal Accuracy (DRMS)
Stand-Alone	Civilian receiver, SA on (historical)	100 m
	Civilian receiver, SA off (current)	5 - 8 m
	Military receiver, (dual frequency)	3 - 5 m
Differential	Code differential	1 - 3 m
	Carrier-smoothed code differential	0.1 - 1 m
	Precise carrier-phase (kinematic)	1 - 2 cm
	Precise carrier-phase (static)	1 - 2 mm

In order to provide the most precise navigation solution possible, particularly in a dynamic environment, a method must be utilized to combine the large number of individual measurements into a meaningful output. One means of combining measurements that is quite common in navigation algorithms is the use of a Kalman filter. Developed by R.E. Kalman in 1960, the filter is a recursive solution to the discrete-data linear filtering problem [21]. Through the use of mathematical equations, an extremely efficient means is provided to allow the estimation of the state of a system while minimizing errors.

## 1.2 Problem Definition

The objective behind the research presented in this thesis was to develop an instrumentation package that combines the use of DGPS, the resolution of carrier-phase ambiguities, and Kalman filtering to compute the precise relative position between two small UAV's flying in close formation. In order to ensure success, the system was required to be able to process real time data from two GPS receivers at a rate of 20 Hertz. Additionally, because the GPS receivers were physically located in different UAV's, the data from the lead aircraft had to be transmitted to the wing aircraft by means of a wireless serial data link. Finally, since the goal involved small UAV's, a key objective was to maximize performance while simultaneously minimizing the overall size, weight, and power consumption of the system. Figure 1.1 depicts the overall system architecture used during development and testing.

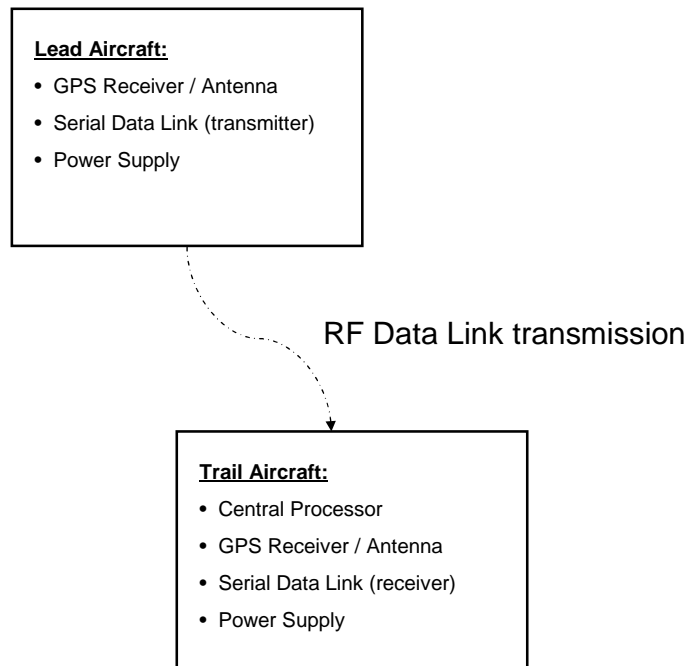


Figure 1.1: System Architecture

### ***1.3 Related Research***

Calculating the precise relative position between two manned aircraft is not a new concept. In 1994, research conducted by Lachapelle, et. al. [7] utilized a pair of U.S. Navy P-3 Orion aircraft. Each aircraft was configured with a pair of NovAtel GPS receivers. One antenna was positioned on the fuselage with the second antenna positioned approximately 20 meters aft on the tail boom. Several of the findings are particularly interesting. First, it was determined that it was possible to detect length variations between the front and rear antenna of each aircraft as small as 1-2 cm. This variation in length, caused by temperature variations, could only be detected after the GPS carrier-phase ambiguities were resolved. Second, because the GPS receivers were single frequency (L1 only), constraint equations were used to simplify the resolution of GPS carrier-phase ambiguities. The constraint equations used were the known distances between the front and rear antennas on each respective aircraft. Finally, during flight tests it was determined that the accuracy of the three dimensional relative position vector between the two aircraft was also on the centimeter level.

The system presented in this thesis utilizes DGPS. However, because each UAV contains only a single GPS receiver, it does not use constraint equations. Instead, dual frequency receivers are used to simplify the resolution of carrier-phase ambiguities.

In 2000, Williamson conducted research on real time, high accuracy, relative state estimation for multiple vehicle systems. In addition to the GPS, wireless communication, and computer support, Williamson's system included an inertial measurement unit (IMU). The results obtained included a relative range estimation error of less than 5 cm, relative roll and pitch less than 0.2 degrees, and relative yaw less than 0.7 degrees. Of note, the system was tested on F-18's at the NASA Dryden Flight Research Center. It was determined that the drag reduction obtained by maintaining close formation and aligning aircraft vortices was approximately 15% [22].

Because of the desire to minimize size, weight, and power consumption for incorporation into small UAV's, the system presented in this thesis does not include

an IMU. However, the output of the system is 20 Hertz which is double the 10 Hertz output obtained in the UCLA research by Williamson.

Research conducted by Spinelli [18] at the Air Force Institute of Technology (AFIT) demonstrated the feasibility of a Real Time Kinetic (RTK) DGPS precise relative positioning system for formation flight between manned aircraft. The system utilized a Javad JNS100 GPS receiver in each aircraft. Although these receivers were single channel, they were selected because they were capable of providing raw GPS measurements at a 100 Hertz rate. An extended Kalman filter was utilized to provide an approximate relative position and a floating point carrier-phase ambiguity estimate. These floating point ambiguities were then fed into an ambiguity resolution routine utilizing the LAMBDA method. A bank of 50 possible ambiguity sets were stored for use in the Multiple Model Adaptive Estimator (MMAE) filters which provide a weighted relative position solution. Because dual frequency measurements were not available, the system required a two minute convergence period to adequately determine the carrier-phase ambiguities. Although the next generation system described in this thesis utilizes dual frequency GPS receivers, and does not use MMAE filters, the remainder of the system is similar in functionality to the system developed by Spinelli. As such, several of his findings worthy of discussion. First, although the research involved single frequency GPS receivers, it was determined that dual frequency GPS receivers are more advantageous. The computational overhead involved in terms of initialization is drastically reduced when both L1 and L2 signals are available. Second, the serial data link used in the research was determined to be incapable of transmitting the necessary data at a 100 Hz rate. Because most flight control algorithms operate at a 100 Hertz rate, a smoothing algorithm will have to be incorporated if the precise relative position is only calculated at a 20 Hertz rate. Third, it was determined that the Windows operating system was unusable for RTK algorithms operating at a 20 Hertz rate. This is due to the numerous processes that are inherently running in the background which cannot be easily de-prioritized. As such, development was conducted in a UNIX environment. Finally, the two aircraft



used for testing were manned aircraft. This resulted in a readily available power supply and very few size and weight restrictions.

#### **1.4 Assumptions**

As a result of the findings mentioned above, several key assumptions were made at the beginning of the current research effort. It was assumed that both GPS receivers would be capable of providing dual frequency measurements. The process of resolving carrier-phase ambiguities is significantly more time consuming if dual frequency measurements are not available. Additionally, even though flight control logic generally operates at a 100 Hertz rate, the decision was made to use GPS receivers that had a maximum output rate of only 20 Hertz. This decision was made because of the inherent limitations of baud rate when using a wireless serial data link. It is therefore assumed that a time averaging method will be employed to smooth the 20 Hertz output prior to it being processed by an autopilot operating at 100 Hertz. Lastly, because of the assumption that Windows would be unusable in a RTK environment, development took place in a Unix / C++ environment.

#### **1.5 Methodology**

The first stage of the research focused on gaining an understanding of the process of calculating the precise position from raw GPS code and carrier-phase measurements. The second stage research involved additional software development to fully understand the added complexity that occurs when data has to be processed in real-time as opposed to using post-processed data. The next stage in the spiral development cycle involved transporting the software processes from a Windows XP / MATLAB environment to a Linux / C++ environment. This transition to UNIX was made because previously conducted research indicated that Windows was not suitable for real-time processing at a 20 Hertz rate. The final stage involved optimizing the hardware configuration and conducting both static and dynamic ground tests.

## ***1.6 Thesis Overview***

The material in Chapter 2 provides the theoretical background necessary to fully understand the topics in later chapters. Primary topics discussed include general GPS theory, GPS carrier-phase ambiguity resolution, and Kalman filtering. Additionally, a description of the hardware components utilized during development and testing is provided. Chapter 3 describes the development cycle of the instrumentation package. Results of both static and dynamic ground tests are presented in Chapter 4. Unfortunately, time constraints did not allow for the completion of dynamic flight tests. Chapter 5 summarizes the results and provides some recommendations for future research efforts in this area.

## II. Background

### 2.1 Overview

This chapter describes the theoretical background for the material presented in later chapters. The first section provides an overview of general GPS theory. A more detailed discussion on the resolution of carrier-phase ambiguities is provided in the second section. The third section describes the fundamentals of Kalman filtering. The chapter concludes with a description of the hardware components utilized during the development and testing of the instrumentation package.

### 2.2 General GPS Theory

The topic of GPS is extremely broad in nature and several excellent references [9,13,14] are available depending on one's specific area of interest. However, the topics discussed in this section are limited in scope to those which must be understood prior to reading the material in subsequent chapters.

*2.2.1 Basic Principles.* For thousands of years, man relied upon the natural stars and angular measurements to be able to compute his position. In a very general sense, given three angular measurements and an almanac which provides a star's position at a given time, one can triangulate their position relatively easily. However, the ability to compute this position is predicated upon one's ability to have a clear view of the stars. This obviously rules out navigation solutions during daylight and/or inclement weather.

With the advent of GPS, the use of natural stars for navigation was replaced by a satellite based radio navigation system thereby providing a means to navigate at anytime in virtually any weather condition. A man-made constellation was created which consists of at least 24 Satellite Vehicles (SVs) placed in six orbital planes (4 SVs in each orbital plane). For redundancy, the number of SVs in orbit at any given time is usually higher, but never exceeds 32.

A key difference between GPS and the use of the natural stars is that GPS does not utilize angular measurements to compute a navigation solution via triangulation. Instead, GPS uses the distances between the SVs and the GPS receiver to compute a navigation solution via trilateration. This is because it is much easier to precisely determine the distance between the SV and the GPS receiver than it is to determine the precise bearing and elevation of arrival of the incoming signal.

In theory, if every GPS receiver contained an atomic clock that was perfectly synchronized to the atomic clocks onboard the SVs, one would only need three SVs to be able to achieve a 3-D navigation solution. However, to avoid such a stringent requirement, a minimum of four SVs must be tracked in order to solve for the four unknowns of position (X,Y,Z components) and receiver clock error. This allows GPS receivers to be much more affordable by allowing the use of an inexpensive quartz clock instead of an atomic clock. Since the receiver clock error will be consistent for all simultaneous measurements in a receiver, it can be estimated, just like the three components of position, given a minimum of four visible SVs.

Thus, in order to compute a navigation solution through the use of GPS, all that is required, from the user's perspective, is a GPS receiver/antenna combination that has a clear line of sight to at least 4 SVs. By calculating the distances between the receiver and each respective SV, the unknowns of position (X,Y,Z components) and clock error can be simultaneously determined.

*2.2.2 GPS Measurements.* The remainder of this section follows Bouska in both form and content [2]. The GPS ranging signal is simultaneously broadcast on two frequencies. These frequencies are referred to as  $L1$  (1575.42 MHz) and  $L2$  (1227.60 MHz). Although each of these two frequencies are capable of having two simultaneous modulations, referred to as phase quadrature,  $L1$  is currently the only frequency with two modulations. The first modulation, referred to as C/A (Coarse Acquisition) Code, consists of a short PRN code broadcast that has a chipping rate of 1.023 MHz. As the C/A Code repeats roughly every millisecond, it is relatively easy to

lock onto. The C/A Code is used primarily by civilian users and is always broadcast in the clear (not encrypted). The second modulation, referred to as Precision (P) Code, consists of a much longer PRN code that has a much faster chipping rate of 10.23 MHz. Because the P Code repeats roughly once a week, GPS receivers generally acquire the C/A code first and then lock onto the P code. The P Code is intended to be exclusively used by the military and is therefore encrypted. Both L1 and L2 are capable of transmitting C/A code and P code simultaneously.

Regardless of whether L1 C/A code, L1 P code, or L2 P code are selected, there are two types of measurements that can be used to determine the distance between the GPS receiver and the SV. These are referred to as Code measurements and Carrier-Phase measurements. Both types of measurements provide range information. The difference between the two lies in the comparison between precision and accuracy.

*2.2.2.1 Code Measurements.* Because the wavelength of the C/A code is approximately 290 meters, compared to only 0.2 meters for the carrier-phase signal, the resulting code measurements are not as precise as the carrier-phase measurements. However, the range measurements obtained are known absolutely and can be expressed in meters. The pseudorange is nothing more than the time difference between transmit time and receive time, multiplied by the speed of light. In contrast, the carrier-phase measurements, discussed in the next section, are expressed in cycles and contain an ambiguity term that must be determined before the measurements can be used.

The pseudorange measurement (in meters) can be expressed as:

$$\rho = r + c(\delta t_r - \delta t_{sv}) + T + I + m_\rho + v_\rho \quad (2.1)$$

where

$\rho \equiv$  GPS pseudorange measurement (meters)

$r \equiv$  actual range from receiver to satellite (meters)

$c \equiv$  speed of light (meters / second)

$\delta t_r \equiv$  receiver clock error (seconds)

$\delta t_{sv} \equiv$  satellite clock error (seconds)

$T \equiv$  tropospheric error (meters)

$I \equiv$  ionospheric error (meters)

$m_\rho \equiv$  multipath error of pseudorange measurement (meters)

$v_\rho \equiv$  noise error of pseudorange measurement (meters)

#### 2.2.2.2 Carrier-Phase Measurements.

Carrier-phase measurements are extremely precise but ambiguous. This is because the wavelength of the carrier-phase signal is approximately 0.2 meters compared to the wavelength of the C/A code which is approximately 290 meters. However, carrier-phase measurements are not known absolutely. Instead, they are ambiguous in nature and are expressed in cycles. The receiver can keep track of the carrier-phase shift quite easily. However, it does not know a priori the number of integer cycles that were present at the time the accumulation began. This is referred to as the carrier-phase ambiguity. In order for carrier-phase measurements to be both precise and accurate, this ambiguity must be resolved to its actual integer value using one of several methods that have been developed over the years. A more detailed description of GPS carrier-phase ambiguity resolution is provided later in this chapter.

The carrier-phase measurement (in cycles) is markedly similar to the preceding equation for pseudorange.

$$\phi = \lambda^{-1}(r + c(\delta t_r - \delta t_{sv}) + T - I + m_\phi + v_\phi) + N \quad (2.2)$$

where

$\phi \equiv$  GPS carrier-phase measurement (cycles)

$\lambda \equiv$  carrier-phase wavelength (meters / cycle)

$m_\phi \equiv$  multipath error of carrier-phase measurement (meters)

$v_\phi \equiv$  noise error of carrier-phase measurement (meters)

$N \equiv$  carrier-phase integer ambiguity (cycles)

The remaining terms in the above equation are as defined in Equation (2.1). It should be noted that the signs of the ionospheric error terms are reversed between the two equations. This is due to a phenomena known as code-carrier divergence in which the ionosphere delays the pseudorange measurement while advancing the carrier-phase measurement.

The last term in Equation (2.2) represents the carrier-phase integer ambiguity and will be discussed in much more detail in the next section. However, for the time being, one can look at this term as an initially unknown bias that is added to each carrier-phase measurement [15]. This is because the GPS receiver can only track the accumulated doppler beginning from an initial time epoch and has no way of knowing the number of cycles between the satellite and the receiver at the instant tracking began.

*2.2.3 Single Differencing.* By computing the measurement differences between two receivers and a common satellite, the satellite clock error ( $\delta t_{sv}$ ) can be eliminated and the atmospheric errors ( $T$  and  $I$ ) can be reduced. Figure 2.1 is used to demonstrate the calculation of a single-difference carrier-phase measurement. In the discussions to follow, superscripts are used to represent the satellite and subscripts are used to represent the receiver. For example  $\rho_A^k$  is the pseudorange between satellite  $k$  and receiver  $A$ .

The single-difference carrier-phase measurement between two receivers and a common SV is defined as:

$$\Delta\phi_{AB}^k = \phi_A^k - \phi_B^k \quad (2.3)$$

where

$\phi_A^k \equiv$  Phase measurement between receiver A and satellite k

$\phi_B^k \equiv$  Phase measurement between receiver B and satellite k

The cancellation of the SV clock error can be seen by substituting Equation (2.2) into Equation (2.3) as follows:

$$\begin{aligned} \Delta\phi_{AB}^k = & (\lambda^{-1}(r_A^k + c(\delta t_{r_A} - \delta t_{sv}^k) + T_A^k - I_A^k + m_{\phi_A}^k + v_{\phi_A}^k) + N_A^k) \\ & - (\lambda^{-1}(r_B^k + c(\delta t_{r_B} - \delta t_{sv}^k) + T_B^k - I_B^k + m_{\phi_B}^k + v_{\phi_B}^k) + N_B^k) \end{aligned} \quad (2.4)$$

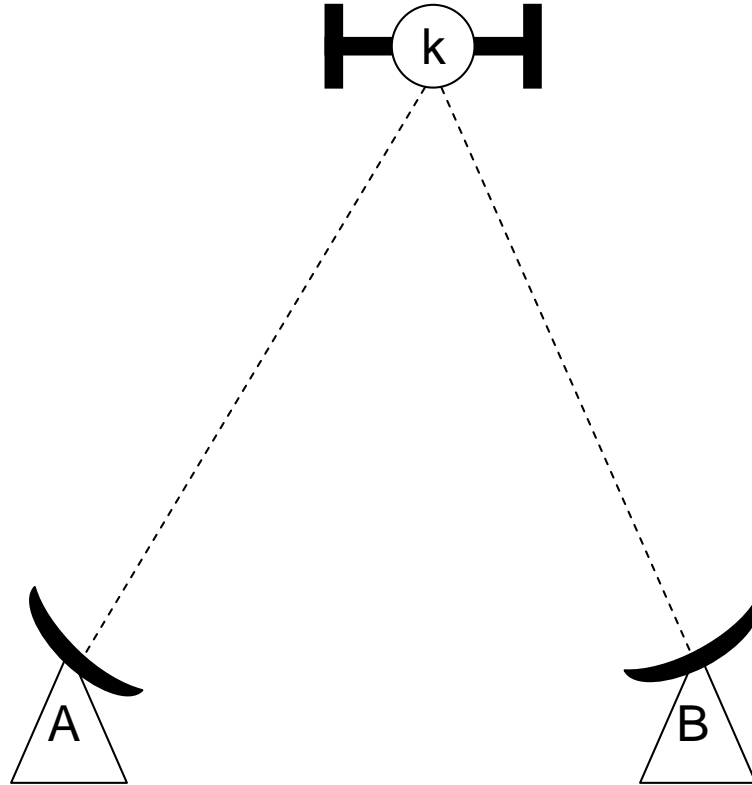


Figure 2.1: Single-Difference Measurement: Two receivers and one SV



Combining like terms and simplifying yields the following:

$$\Delta\phi_{AB}^k = \lambda^{-1}(\Delta r_{AB}^k + c\Delta\delta t_{r_{AB}} + \Delta T_{AB}^k - \Delta I_{AB}^k + \Delta m_{\phi_A\phi_B}^k + v_{\phi_A\phi_B}^k) + \Delta N_{AB}^k \quad (2.5)$$

The last term in Equation (2.5),  $(\Delta N_{AB}^k)$ , is referred to as the single-difference carrier-phase ambiguity.

*2.2.4 Double Differencing.* A double-difference measurement is nothing more than the difference between two single-difference measurements. The utility of such a measurement becomes apparent when one forms a single-difference between a satellite and a pair of receivers (as discussed in the preceding section) and another single-difference between a receiver and a pair of satellites. The first single-difference eliminates the receiver clock error while the second single-difference eliminates the satellite clock error. The newly formed double-difference effectively eliminates both satellite clock error and receiver clock error. Figure 2.2 is used to demonstrate the calculation of a double-difference carrier-phase measurement.

Using the same notation as in the single-difference case, the double-difference carrier-phase measurement is defined as follows:

$$\Delta\nabla\phi_{AB}^{jk} = \Delta\phi_{AB}^j - \Delta\phi_{AB}^k \quad (2.6)$$

The cancellation of the receiver clock error can be seen by substituting Equation (2.5) into Equation (2.6) as follows:

$$\begin{aligned} \Delta\nabla\phi_{AB}^{jk} &= \lambda^{-1}(\Delta r_{AB}^j + c\Delta\delta t_{r_{AB}} + \Delta T_{AB}^j - \Delta I_{AB}^j + \Delta m_{\phi_A\phi_B}^j + v_{\phi_A\phi_B}^j) + \Delta N_{AB}^j \\ &\quad - \lambda^{-1}(\Delta r_{AB}^k + c\Delta\delta t_{r_{AB}} + \Delta T_{AB}^k - \Delta I_{AB}^k + \Delta m_{\phi_A\phi_B}^k + v_{\phi_A\phi_B}^k) + \Delta N_{AB}^k \end{aligned} \quad (2.7)$$

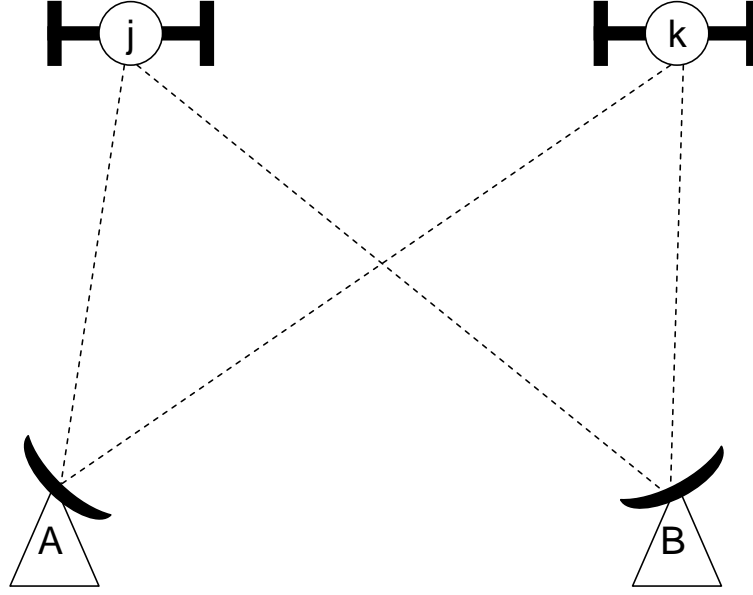


Figure 2.2: Double-Difference Measurement: Two receivers and two SVs

Combining like terms and simplifying yields:

$$\Delta\nabla\phi_{AB}^{jk} = \lambda^{-1}(\Delta\nabla r_{AB}^{jk} + \Delta\nabla T_{AB}^{jk} - \Delta\nabla I_{AB}^{jk} + \Delta\nabla m_{AB}^{jk} + \Delta\nabla v_{AB}^{jk}) + \Delta\nabla N_{AB}^{jk} \quad (2.8)$$

The last term in Equation (2.8),  $\Delta\nabla N_{AB}^{jk}$ , is referred to as the double-difference carrier-phase ambiguity.

Of note, although the preceding discussion was for single and double-difference carrier-phase measurements, the same methodology can be utilized to calculate single and double-difference code measurements.

*2.2.5 Virtual Measurements.* Because the GPS signal is transmitted on two different frequencies, it is possible to use linear combinations of the raw measurements to formulate new measurements. The most common linear combination utilized in DGPS applications is frequently referred to as widelane. For carrier-phase measurements, the new widelane measurement is defined as follows:

$$\phi_{WL} \equiv \phi_{L1} - \phi_{L2} \quad (2.9)$$

There are several advantages to using widelane measurements. The wavelength of the widelane measurements is roughly 86.19 cm compared to 19.03 cm for L1 and 24.42 cm for L2. The longer wavelength results in a smaller search space and greatly aids in the carrier-phase ambiguity resolution which is further discussed in the next section.

Another linear combination utilized in DGPS applications is frequently referred to as narrowlane. For carrier-phase measurements, the new narrowlane measurement is defined as follows:

$$\phi_{NL} = \phi_{L1} + \phi_{L2} \quad (2.10)$$

The advantage of using narrowlane, as the name implies, is that the shorter wavelength (approximately 10 cm) results in higher precision. However, because carrier-phase ambiguity is much more difficult in narrowlane, DGPS applications generally start with widelane measurements, transition to single frequency measurements, and then transition to narrowlane measurements.

### ***2.3 Carrier-Phase Ambiguity Resolution***

To obtain the most precise navigation solution possible, the carrier-phase ambiguity terms must be resolved to their integer values. Numerous methods have been developed over the years to perform this ambiguity resolution. Discussion in this section is limited in scope to the method utilized in the development of the system presented in this thesis.

*2.3.1 Basic Principles.* Although there are numerous techniques to perform ambiguity resolution, the majority of the algorithms perform two basic tasks. First, a determination is made regarding the ambiguity search space. This involves creating a number of possible ambiguity sets to be evaluated. The first step is critical to the success of the algorithm. If the collection of ambiguity sets is too small then it may not contain the correct set. However, if it is too large, then it becomes computationally difficult to select the proper set. Once the ambiguity search space

has been determined, the second task is to select the correct ambiguity set and discard the rest.

*2.3.2 The LAMBDA Method.* One of the methods developed to resolve carrier-phase ambiguities to their integer values is the Least-squares AMBiguity Decorrelation Adjustment (LAMBDA) method. Developed by Teunissen, Jonge, and Tiberius [5], the algorithm utilizes a two step process to determine the integer ambiguity values. As the name implies, the first step decorrelates the ambiguities through the use of a Z-transformation. This greatly reduces the size of the ambiguity search space while ensuring the correct solution is still included in the set of possibilities. The second step solves the integer ambiguities by performing a discrete search of the ellipsoidal search space generated by the Z-transformation. The LAMBDA method provides an extremely efficient way of determining the correct ambiguity set. On a relatively slow computer such as a 486-66 MHz PC, the complete process of performing the Z-transformation and determining the correct ambiguity set typically takes less than 30 ms for a baseline of 12 ambiguities [5].

Source code is openly available through Delft University in both MATLAB and FORTRAN formats [6]. The system presented in this thesis utilizes a C++ implementation of the LAMBDA method, designed by the author, which closely follows the MATLAB code mentioned above.

As discussed by Teunissen [19], the first step in the implementation of the LAMBDA method is to perform a Z-transformation. In general, the ambiguity estimates are highly correlated resulting in a covariance matrix where the diagonal terms are not significantly larger than the off-diagonal terms. This condition makes the ambiguity search more difficult. The goal of the Z-transformation is to decorrelate the original ambiguities which makes the search more efficient. The original double-difference carrier-phase ambiguities can be transformed as follows:

$$\mathbf{z} = \mathbf{Z}^T \mathbf{a} \quad (2.11)$$

where

$\mathbf{z} \equiv$  transformed ambiguity vector

$\mathbf{Z}^T \equiv$  Transpose of  $\mathbf{Z}$ -transformation matrix

$\mathbf{a} \equiv$  original ambiguity vector

Similarly, the least-squares estimate of the double-difference carrier-phase ambiguities can be transformed as follows:

$$\hat{\mathbf{z}} = \mathbf{Z}^T \hat{\mathbf{a}} \quad (2.12)$$

where

$\hat{\mathbf{z}} \equiv$  transformed ambiguity estimate

$\hat{\mathbf{a}} \equiv$  original ambiguity estimate

Finally, the double-difference carrier-phase covariance matrix can be transformed as follows:

$$\mathbf{Q}_{\hat{\mathbf{z}}} = \mathbf{Z}^T \mathbf{Q}_{\hat{\mathbf{a}}} \mathbf{Z} \quad (2.13)$$

where

$\mathbf{Q}_{\hat{\mathbf{z}}} \equiv$  transformed covariance matrix

$\mathbf{Q}_{\hat{\mathbf{a}}} \equiv$  original covariance matrix

It is important to note that the  $\mathbf{Z}$ -transformation matrix is required to be volume preserving [19]. Additionally, it must be composed entirely of integer numbers and it must reduce the product of ambiguity variances.

An example of the transformation of the covariance matrix is provided below [17]:

$$\mathbf{Q}_{\hat{\mathbf{a}}} = \begin{bmatrix} 6.290 & 5.978 & 0.544 \\ 5.978 & 6.292 & 2.340 \\ 0.544 & 2.340 & 6.288 \end{bmatrix} \quad (2.14)$$

$$\mathbf{Z}^T = \begin{bmatrix} 1 & -1 & 0 \\ -2 & 3 & -1 \\ 3 & -3 & 1 \end{bmatrix} \quad (2.15)$$

Substituting Equation (2.14) and Equation (2.15) into Equation (2.13) yields the following:

$$\mathbf{Q}_{\hat{\mathbf{z}}} = \begin{bmatrix} 0.626 & 0.230 & 0.082 \\ 0.230 & 4.476 & 0.334 \\ 0.082 & 0.334 & 1.146 \end{bmatrix} \quad (2.16)$$

One can see quite easily that the covariance matrix in Equation (2.16) is significantly more de-correlated than the covariance matrix in Equation (2.14).

For additional information regarding the LAMBDA method, the reader is referred to several excellent sources of information [5,6,19].

## 2.4 *Kalman Filtering*

Introduced in 1960 [21], the Kalman filter has frequently been utilized in the field of navigation as a means to smooth out the inherently noisy nature of position related measurements. Zarchan and Musoff [23], as well as Maybeck [8], provide an excellent description of the fundamentals of Kalman filtering along with the associated mathematical details necessary for implementation. This section, is limited in scope to those aspects of the Kalman filter that must be understood prior to the discussions in following chapters.

*2.4.1 Basic Principles.* In essence, the Kalman filter is used as a means to provide an estimation of the state of a system. By comparing the current state estimate, which is based on all previous measurements, with the noisy measurement data, the filter attempts to provide the most accurate state estimate possible. In order to utilize a Kalman filter, there are two criteria that must be met. First, the system to be estimated must be described by a linear model. Second, all noise processes must be zero-mean, white-Gaussian in nature. If one, or both, of these conditions are not met, the performance of the filter will be adversely impacted. The Kalman filter performs two basic operations [21]. First, the time update, also known as prediction, projects the current state and error covariance estimates forward in time to obtain the estimates for the next time epoch. Second, the measurement update, also known as the correction, is responsible for comparing the estimated measurement with the actual measurement and updating the state to achieve a more precise estimate for the next time epoch.

*2.4.2 Kalman Filter Equations.* In order to utilize a Kalman filter to perform state estimation, the state of the system may be described in the following continuous state-space form:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} + \mathbf{w} \quad (2.17)$$

where

$\mathbf{x}$  is a column vector containing the  $n$  states of the system (dimensions  $n \times 1$ ).

$\mathbf{F}$  is the system dynamics matrix (dimensions  $n \times n$ ).

$\mathbf{G}$  is the control input matrix (dimensions  $n \times r$ ).

$\mathbf{u}$  is the control vector (dimensions  $r \times 1$ ).

$\mathbf{w}$  is a white-noise process (dimensions  $n \times 1$ )

Of note, for the research presented in this thesis, there is no control input. The

$\mathbf{G}$  matrix is included here for completeness only. Since it is zero, it will be left out of all future equations.

The measurements must be linearly related to the states as follows:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (2.18)$$

where

$\mathbf{z}$  is the measurement vector.

$\mathbf{H}$  is the measurement matrix.

$\mathbf{v}$  is the white measurement noise.

Because the GPS measurements are not continuous in nature, and occur at discrete time intervals, the preceding equations are normally converted to a discrete-time case. To do so, the fundamental matrix (also referred to as the transition matrix),  $\Phi$ , is computed as:

$$\Phi(\mathbf{T}_s) \equiv \mathbf{e}^{\mathbf{F}\mathbf{T}_s} \quad (2.19)$$

where

$\mathbf{T}_s$  is the measurement sampling period.



The discrete state vector time update (prediction) is then computed as follows:

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k \quad (2.20)$$

where

$\mathbf{x}_{k+1}$  is the state vector at time  $k+1$

$\Phi_k$  is the fundamental matrix

$\mathbf{x}_k$  is the state vector at time  $k$

Likewise, the state covariance matrix time update (prediction) is computed as follows:

$$\mathbf{P}_{k+1} = \Phi_k \mathbf{P}_k \Phi_k^T + \mathbf{Q} \mathbf{d} \quad (2.21)$$

where

$\mathbf{P}_{k+1}$  is the state covariance at time  $k+1$

$\mathbf{P}_k$  is the state covariance at time  $k$

$\mathbf{Q} \mathbf{d}$  is the discrete noise covariance

There are two different methods that can be utilized to model the noise in the system. The first, known as the random walk, is used to represent a discrete stochastic process in which there is no predisposition for the process to tend towards a given direction and the covariance grows in an unbounded manner. In contrast, the second method, known as the Gauss-Markov process [8], is used to represent a discrete stochastic process in which the covariance is bounded over time. As discussed in Chapter 4, the system presented in this thesis utilizes both of these methods.

The discrete measurement vector  $\mathbf{z}_k$  is modeled as follows:

$$\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k \quad (2.22)$$

where

$\mathbf{H}$  is the measurement matrix.

$\mathbf{v}_k$  is discrete white measurement noise.

Before the discrete measurement vector can be used to update the state vector and covariance matrix, a Kalman gain matrix must be computed as follows:

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}^T (\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R})^{-1} \quad (2.23)$$

where

$\mathbf{K}_k$  is the Kalman gain matrix at time  $k$ .

$\mathbf{P}_k$  is the state vector covariance matrix at time  $k$ .

$\mathbf{H}$  is the measurement matrix.

$\mathbf{R}$  is the measurement noise covariance matrix.

After the new measurements are received, the measurement update (correction) is applied and the new discrete state vector is computed as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \mathbf{x}_k) \quad (2.24)$$

where

$\mathbf{x}_{k+1}$  is the state vector at time  $k+1$ .

$\mathbf{x}_k$  is the state vector at time  $k$ .

$\mathbf{K}_k$  is the Kalman gain matrix at time  $k$ .

$\mathbf{z}_k$  is the measurement at time  $k$ .

$\mathbf{H}$  is the measurement matrix  $k$ .

Similarly, the measurement update (correction) results in a new discrete state covariance matrix as follows:

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{K}_k \mathbf{H} \mathbf{P}_k \quad (2.25)$$

Finally, it must be noted that the Kalman filter requires initial values for both the state vector and the state covariance matrix. Discussions relating to filter initialization will be provided in Chapter 3. For a more in-depth discussion of Kalman filtering, the reader is referred to several excellent sources of information [8, 21, 23].

## 2.5 Hardware Components

The hardware components for the instrumentation package were specifically chosen with the objective of minimizing size, weight, and power consumption requirements. The major components will now be briefly discussed.

*2.5.1 GPS Receiver.* The GPS cards selected for use in the instrumentation package were NovAtel OEM4-G2L's which are depicted in Figure 2.3 [11].

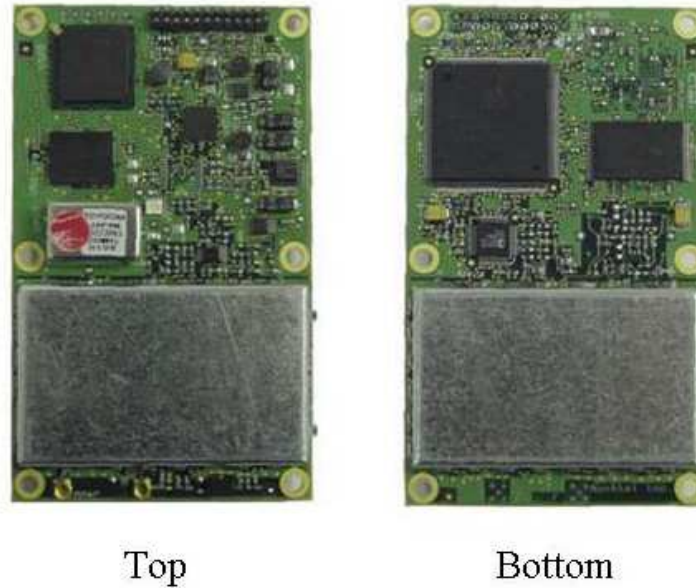


Figure 2.3: NovAtel OEM4-G2L GPS Receiver Card [11]

The OEM4-G2L GPS receiver cards are capable of providing dual frequency, raw GPS measurements at a 20 Hertz rate. Additional features which support the requirements to minimize size, weight, and power consumption are shown in Table 2.1.

Table 2.1: NovAtel OEM4-G2L Specifications [11]

size	6 cm x 10 cm x 1.6 cm
weight	56 grams
typical power consumption	1.6 Watts

Of note, the OEM4-G2L's are 40% smaller than the OEM4-G2's and consume 15% less power.

The NovAtel OEM4-G2L's provide measurements and data in the form of logs. Three separate logs were utilized to provide input to the processing algorithm discussed in the next chapter. The Best Position log (BESTPOS) provides latitude, longitude, and height above sea-level. The Decoded GPS Ephemerides log (GPSEPHM) provides a set of GPS ephemeris parameters which is required to determine the position and orbital information for each SV. The Compressed version of the Range log (RANGECMP) provides raw GPS pseudoranges and carrier-phase measurements. Additionally, the RANGECMP log provides the signal lock time which can be used for cycle slip detection. Cycle slips are discussed in Chapter 3.

*2.5.2 Antenna.* The GPS antennas selected for initial ground testing were Ashtech 700936-E, dual frequency, chokering antennas (with radome). After these antennas were positioned on the roof of the Air Force Institute of Technology, a 24 hour data collection effort (15 second interval) was undertaken. The data was then uploaded to the Online Positioning User Service (OPUS) and a precise position was returned. The difference between the two known positions was calculated to yield the truth value for static ground tests.

The GPS antennas selected for use in the radio control airplanes were ANTCOM 3G1215X1000, dual frequency, active antennas and are shown in Figure 2.4 [10].



Figure 2.4: ANTCOM 3G1215X1000 40 dB GPS Antenna [10]

Overall dimensions and weight are provided in Table 2.2 [1]

Table 2.2: ANTCOM 3G1215X1000 Specifications [1]

diameter	8.9 cm
height	1.6 cm
weight	0.142 grams

*2.5.3 Serial Data Link.* During development, as well as initial static ground testing, the serial data link was not used. Instead, each of the NovAtel GPS receivers was connected to the CPU via a separate serial cable.

The Serial Data Link selected for use during dynamic ground testing, and future incorporation into the radio controlled aircraft, is the Freewave FGR09CS Spread Spectrum Radio Modem shown in Figure 2.5 [4]. Overall dimensions, weight, and power consumption rates are provided in Table 2.3 [4]. Previous testing by Spinelli [18] determined that the Freewave modem could reliably handle data transfer at a 20 Hertz rate. However, the modem did not have the bandwidth to process data at 100 Hertz. For this reason, the NovAtel receivers were chosen since they provide dual frequency measurements at a 20 Hertz rate. Assuming that the NovAtel RANGECMPB data



Figure 2.5: Freewave FGR09CS Wireless Modem [4]

log is providing dual frequency measurements for 12 SVs at a 20 Hertz rate, the required bandwidth of 97.92 Kbps is moderately below the maximum bandwidth of the Freewave FGR09CS which is 115.2 Kbps. This allows for additional data logs such as GPSEPHEMB updates. Of note, to process the data at a rate of 100 Hertz, the bandwidth of the serial data link would have to be increased by a factor of slightly more than four.

Table 2.3: FreeWave FGR09CS Specifications [4]

size	12.7 cm x 6.1 cm x 1.8 cm
weight	74.4 grams
typical power consumption	6 W (transmit), 1 W (receive)

*2.5.4 CPU / Power supply.* Because time did not permit the software to be ported over to PC-104 or GUMSTIX PC boards, all testing was done on a 1.4 GHz desktop computer with a clock speed of 400 Hz. However, the available processing power on select PC-104 and/or GUMSTIX PC boards meets or exceeds the system on which the instrumentation package was tested on. Additionally, in the case of the GUMSTIX PC boards, the LINUX operating system is burned into the read-only memory. Lastly, both PC-104 and GUMSTIX systems support multi-threading.

## 2.6 Summary

This chapter has provided the reader with the necessary background to be able to follow the discussion in later chapters. In the material to follow, general GPS

theory, resolution of carrier-phase ambiguities, and Kalman filtering will be combined with the goal of developing a system whose primary algorithm rapidly calculates the precise relative position between miniature UAV's while minimizing the system size, weight and power consumption.

### III. Development of Instrumentation Package

#### 3.1 Overview

This chapter describes in detail the development of an instrumentation package which provides a precise relative position vector between two UAV's that is suitable for formation flight control. After providing a big picture overview of the top-level design, some of the factors that must be considered when designing real-time DGPS systems are discussed. After a brief description of the front-end parser used to synchronize the raw measurements, the remainder of the chapter discusses the three main tasks that are performed at various discrete time intervals. These consist of a Kalman filter task operating at a 1 Hertz rate, a LAMBDA ambiguity resolution task which is called by the Kalman filter when required, and a high-rate precise relative position output task operating at a 20 Hertz rate.

#### 3.2 Top-Level Design

The instrumentation package discussed in this chapter was developed to provide a high-rate, low-latency, extremely precise relative position vector between a lead UAV and a wing UAV flying in close formation. Additionally, because the system was designed for small UAV's, emphasis was placed on minimizing the size, weight, and power consumption of the system. Figure 3.1 is used to assist in the visualization of the interaction of the various tasks which are discussed below.

In order to compute the relative position vector, raw GPS measurements are provided by a single NovAtel GPS card in each aircraft. The lead aircraft broadcasts these measurements to the wing aircraft through the use of a Freewave serial data link wireless communication device. In the case of the wing aircraft, the NovAtel GPS card is connected directly to the CPU by means of a serial cable. Once the system begins receiving data from both aircraft, a front-end parser is utilized to synchronize the data into measurements which are common to both aircraft. Once these common measurements have passed through the parser, there are three core tasks, discussed below, which operate asynchronously yet share information when needed.



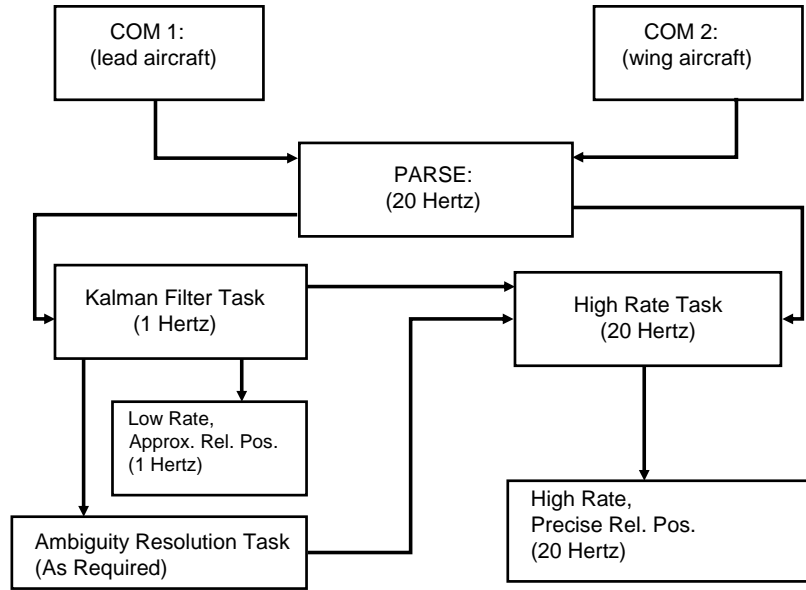


Figure 3.1: Top Level Design of Instrumentation Package

The first task, performed at a 1 Hertz rate, is a floating point Kalman filter. The state vector in the Kalman filter contains the floating point double-difference carrier-phase ambiguity estimates. The state vector contains additional data which is dependent on the operating mode selected by the user. In position mode, the first three rows of the state vector contain the relative position of the wing aircraft with respect to the lead aircraft. In position-velocity mode, the first six rows of the state vector contain the relative position and relative velocity respectively of the wing aircraft with respect to the lead aircraft. In position-velocity-acceleration mode, the first nine rows of the state vector contain the relative position, relative velocity, and relative acceleration respectively of the wing aircraft with respect to the lead aircraft. The floating point double-difference carrier-phase ambiguity estimates start at row four, row seven, or row ten of the state vector, depending on the operating mode selected by the user. The Kalman filter provides several useful outputs. First, the floating point carrier-phase ambiguity estimates, as well as the covariance matrix, are fed to the LAMBDA ambiguity resolution task. Second, the unit line-of-sight vector to each of the SVs is fed to the high-rate output task. Finally, the Kalman filter provides a low rate (1 Hertz), approximate relative position which is independent of

the computations performed by the high-rate output task. This provides a means of validating the high-rate output since a direct comparison is possible.

The second task, an ambiguity resolution routine, is summoned by the Kalman filter when required. The Kalman filter can only provide a floating point estimate of the double-difference carrier-phase ambiguities. As previously discussed, these floating point estimates must be resolved to their integer values in order for the high-rate output to achieve the desired level of precision. The LAMBDA method, mentioned briefly in Chapter 2, is used to perform this ambiguity resolution. The LAMBDA method requires as input the floating point estimates of the ambiguity values as well as the associated covariance matrix. Both of these are provided by the Kalman filter task. The ambiguity resolution task provides as output a vector of the integer ambiguities to the high-rate output task which is discussed next. Of note, after the integer ambiguities have been initially determined, the high-rate task is able to determine the correct integer ambiguities for newly acquired SVs. However, if too many SVs are lost at a single epoch, the Kalman filter has the ability to pass the required data to the ambiguity resolution routine and re-calculate the complete ambiguity set. This will be discussed in much greater detail later in the chapter.

The final task is the low-latency, high-rate, extremely precise, relative position output. This task operates at a 20 Hertz rate and receives information from both the Kalman filter task as well as the LAMBDA ambiguity resolution task. In an ideal case, the high-rate output would have all data available at a 20 Hertz rate. However, this is computationally intensive and is impractical for applications such as miniature UAV's where size, weight, and power consumption requirements are extremely stringent. An in-depth discussion of the effect of using data with a 1 Hertz update rate in a 20 Hertz process will be delayed until Chapter 4.

### ***3.3 Real-Time Considerations***

Implementing an algorithm that operates real-time is significantly more complex than one that utilizes post-processed data. The system presented in this thesis

is capable of operating in both the real-time environment and the post-processed environment. When the system operates in real-time mode, an output file is generated which contains all of the raw GPS measurements. This file can then be used as input when the system is operating in post-processed mode. This allows the same raw GPS measurements to be processed multiple times, with different Kalman filter tuning parameters, and was extremely beneficial during development and testing.

When utilizing DGPS, the only measurements that may be used are from those SVs that are common to both the reference receiver and the mobile receiver. If one of the receivers has data from a SV that the other does not see, the data from the former is essentially useless and must be discarded. The system presented in this thesis utilizes a front-end parser, discussed in a later section, to synchronize the raw GPS measurements from the two receivers. When operating in real-time mode, the parser sends data to the other processes and also generates a data file consisting of the synchronized raw measurements which can then be used as inputs when operating in post-processed mode. It is important to note that the parser does not account for the gain and/or loss of SVs. This is a real-time consideration, discussed below, which must be dynamically handled regardless of whether the user selects real-time mode or post-processed mode.

When a system must operate in a real-time environment, there will be numerous instances of new SVs coming into view and existing SVs falling below the line of sight and becoming unavailable. Additionally, even though the SV may be in view, there may be momentary instances where the GPS signal is temporarily unavailable. These instances, referred to as cycle slips, are generally very short in duration. However, cycle slips must be accounted for regardless of how short in duration they are.

The first question that must be answered when a SV becomes unavailable is whether or not the SV was the base SV used for each of the double-difference pseudo-range and carrier-phase measurements. If an SV, which is not the base SV, becomes unavailable then a single row of the state vector must be eliminated. Additionally, a

single row and a single column must be removed from the covariance matrix. Finally, if the total number of SVs change, then several other matrices require re-sizing so that the matrices in the equations to follow are of the proper dimensions. If the base SV becomes unavailable then the process is more complicated. Because the loss of the base SV affects every double-difference pseudorange and carrier-phase measurement, a transformation matrix must be formed which will allow a new state vector and new covariance matrix to be formed which utilize a new base SV. An example of each of these scenarios will now be provided.

*3.3.1 Loss of Non-Base SV.* For this example, it is assumed that there are 5 SVs in view. Additionally, for simplicity the SVs are numbered sequentially with PRN 1 being the base SV and PRN 3 being the SV that goes out of view. To minimize the size of the matrices, it is assumed that the system is operating in position only mode with the double-difference carrier-phase ambiguity terms starting in the fourth row of the state vector. Prior to the loss of the SV, the state vector and covariance matrix is as follows:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \Delta\nabla\phi^{1,2} \\ \Delta\nabla\phi^{1,3} \\ \Delta\nabla\phi^{1,4} \\ \Delta\nabla\phi^{1,5} \end{bmatrix} \quad (3.1)$$

$$\mathbf{P} = \begin{bmatrix} \sigma_{x,x} & \sigma_{x,y} & \sigma_{x,z} & \dots & \dots & \dots & \dots \\ \sigma_{y,x} & \sigma_{y,y} & \sigma_{y,z} & \dots & \dots & \dots & \dots \\ \sigma_{z,x} & \sigma_{z,y} & \sigma_{z,z} & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{1,2}} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{1,3}} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{1,4}} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{1,5}} \end{bmatrix} \quad (3.2)$$

After accounting for the loss of PRN 3, the new state vector and covariance matrix is as follows:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \Delta\nabla\phi^{1,2} \\ \Delta\nabla\phi^{1,4} \\ \Delta\nabla\phi^{1,5} \end{bmatrix} \quad (3.3)$$

$$\mathbf{P} = \begin{bmatrix} \sigma_{x,x} & \sigma_{x,y} & \sigma_{x,z} & \dots & \dots & \dots \\ \sigma_{y,x} & \sigma_{y,y} & \sigma_{y,z} & \dots & \dots & \dots \\ \sigma_{z,x} & \sigma_{z,y} & \sigma_{z,z} & \dots & \dots & \dots \\ \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{1,2}} & \dots & \dots \\ \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{1,4}} & \dots \\ \dots & \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{1,5}} \end{bmatrix} \quad (3.4)$$

*3.3.2 Loss of Base SV.* For this example, it is also be assumed that there are 5 SVs in view, numbered sequentially. The old base SV is PRN 3, and the new

base SV is PRN 4. It will again be assumed that the system is operating in position only mode with the double-difference carrier-phase ambiguity terms starting in the fourth row of the state vector. Prior to the loss of the base SV, the state vector and covariance matrix is as follows:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \Delta\nabla\phi^{3,1} \\ \Delta\nabla\phi^{3,2} \\ \Delta\nabla\phi^{3,4} \\ \Delta\nabla\phi^{3,5} \end{bmatrix} \quad (3.5)$$

$$\mathbf{P} = \begin{bmatrix} \sigma_{x,x} & \sigma_{x,y} & \sigma_{x,z} & \dots & \dots & \dots & \dots \\ \sigma_{y,x} & \sigma_{y,y} & \sigma_{y,z} & \dots & \dots & \dots & \dots \\ \sigma_{z,x} & \sigma_{z,y} & \sigma_{z,z} & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{3,1}} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{3,2}} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{3,4}} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi^{3,5}} \end{bmatrix} \quad (3.6)$$

Upon inspection of Equation (3.5) and Equation (3.6), one can see that the loss of PRN 3 affects multiple rows. One option that exists is to re-initialize the entire state vector and covariance matrix. However, a better alternative is to create a transformation matrix ( $\mathbf{T}$ ) that allows a switch to a different base SV without losing valuable data. The new state vector and covariance matrix can then be calculated as follows:

$$\mathbf{x}_{\text{new}} = \mathbf{T}\mathbf{x}_{\text{old}} \quad (3.7)$$

$$\mathbf{P}_{\text{new}} = \mathbf{T}\mathbf{P}_{\text{old}}\mathbf{T}^T \quad (3.8)$$

The formation of the  $\mathbf{T}$  matrix in Equation (3.7) and Equation (3.8) is possible because of the fact that the double-difference carrier-phase measurements are linear combinations. For example, consider the equations below:

$$\begin{aligned} \Delta\nabla N^{4,3} &= \Delta N^4 - \Delta N^3 = -(\Delta N^3 - \Delta N^4) \\ &= -\Delta\nabla N^{3,4} \end{aligned} \quad (3.9)$$

$$\begin{aligned} \Delta\nabla N^{4,5} &= \Delta N^4 - \Delta N^5 = \Delta N^4 - \Delta N^3 + \Delta N^3 - \Delta N^5 \\ &= (-\Delta\nabla N^{3,4}) + \Delta\nabla N^{3,5} \end{aligned} \quad (3.10)$$

By utilizing linear combinations such as those in Equation (3.9) and Equation (3.10), a  $\mathbf{T}$  matrix that switches the base SV can be formed for any combination of old and new base SVs. For the example above, where the old base SV of PRN 3 is replaced by the new base SV of PRN 4, the  $\mathbf{T}$  matrix is as follows:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \quad (3.11)$$

Utilizing Equation (3.7), Equation (3.8), and (3.11) yields following state vector and covariance matrix:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \Delta \nabla \phi^{4,1} \\ \Delta \nabla \phi^{4,2} \\ \Delta \nabla \phi^{4,3} \\ \Delta \nabla \phi^{4,5} \end{bmatrix} \quad (3.12)$$

$$\mathbf{P} = \begin{bmatrix} \sigma_{x,x} & \sigma_{x,y} & \sigma_{x,z} & \dots & \dots & \dots & \dots \\ \sigma_{y,x} & \sigma_{y,y} & \sigma_{y,z} & \dots & \dots & \dots & \dots \\ \sigma_{z,x} & \sigma_{z,y} & \sigma_{z,z} & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \sigma_{\Delta \nabla \phi^{4,1}} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \sigma_{\Delta \nabla \phi^{4,2}} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \sigma_{\Delta \nabla \phi^{4,3}} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \sigma_{\Delta \nabla \phi^{4,5}} \end{bmatrix} \quad (3.13)$$

A comparison of Equation (3.5), Equation (3.6), Equation (3.12), and Equation (3.13) reveals that whereas the loss of SV 3 affects numerous rows and columns in the first two equations, it only affects a single row and column in the second two equations. This is a direct result of the use of the transformation matrix. It must be noted that accounting for a change in the base SV is a two-step process. First, the state vector and covariance matrix must be transformed as shown above. After the transformation is performed, the single row containing the old base SV must be removed from the state vector. Similarly, the single row and column containing the old base SV must be removed from the covariance matrix. However, it is important to note that these rows and columns may not be removed until after the transformation is complete. An



additional consideration is that the SV selected as the new base must be present at both the current epoch as well as the previous epoch.

*3.3.3 Gain of SV.* Should an additional SV be acquired, both the state vector and covariance matrix must be expanded to account for the additional measurements. Consider the case where the system originally has the following SVs in view: 1, 3, 5, 7. It is assumed that PRN 1 is the base SV. It will also be assumed that the newly acquired SV is PRN 4. Before acquisition, the state vector and covariance matrix is as follows:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \Delta\nabla\phi_{1,3} \\ \Delta\nabla\phi_{1,5} \\ \Delta\nabla\phi_{1,7} \end{bmatrix} \quad (3.14)$$

$$\mathbf{P} = \begin{bmatrix} \sigma_{x,x} & \sigma_{x,y} & \sigma_{x,z} & \dots & \dots & \dots & \dots \\ \sigma_{y,x} & \sigma_{y,y} & \sigma_{y,z} & \dots & \dots & \dots & \dots \\ \sigma_{z,x} & \sigma_{z,y} & \sigma_{z,z} & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \sigma_{\Delta\nabla\phi_{1,3}} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi_{1,5}} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \sigma_{\Delta\nabla\phi_{1,7}} & \dots \end{bmatrix} \quad (3.15)$$

After accounting for the gain of PRN 4, the new state vector and covariance matrix

is as follows:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ \Delta\nabla\phi_{1,3} \\ \Delta\nabla\phi_{1,4} \\ \Delta\nabla\phi_{1,5} \\ \Delta\nabla\phi_{1,7} \end{bmatrix} \quad (3.16)$$

$$\mathbf{P} = \begin{bmatrix} \sigma_{x,x} & \sigma_{x,y} & \sigma_{x,z} & \dots & 0 & \dots & \dots \\ \sigma_{y,x} & \sigma_{y,y} & \sigma_{y,z} & \dots & 0 & \dots & \dots \\ \sigma_{z,x} & \sigma_{z,y} & \sigma_{z,z} & \dots & 0 & \dots & \dots \\ \dots & \dots & \dots & \sigma_{\Delta\nabla\phi_{1,3}} & 0 & \dots & \dots \\ 0 & 0 & 0 & 0 & \sigma_{\Delta\nabla\phi_{1,4}} & 0 & 0 \\ \dots & \dots & \dots & \dots & 0 & \sigma_{\Delta\nabla\phi_{1,5}} & \dots \\ \dots & \dots & \dots & \dots & 0 & \dots & \sigma_{\Delta\nabla\phi_{1,7}} \end{bmatrix} \quad (3.17)$$

### 3.4 Front-End Parser

The system presented in this thesis utilizes three data logs from the NovAtel GPS receivers. The binary version of these logs was selected for use since they are roughly half the size of the ASCII version. As mentioned in the previous section, any DGPS application requires that the raw measurements be synchronized. After a brief discussion of the data logs, a description of the method utilized to synchronize the raw measurements will be provided.

The NovAtel GPSEPHMEMB data log was selected to provide the orbital ephemeris data for each SV. Because this data is common to both receivers, combined with the fact that the size of the data log is quite large, it is read directly from the GPS card

located in the wing aircraft and is not transmitted by the lead aircraft over the serial data link. The GPS card outputs this information once at initialization and whenever new ephemeris information becomes available.

The NovAtel BESTPOSB data log was selected to provide the GPS receiver computed best position fix. This data log is obtained from both GPS receivers (lead and wing aircraft), and is passed to the CPU at a 1 Hertz rate. This data log is used to initialize the relative position in the state vector. The data for the wing aircraft is also used in the computation of the unit line-of-sight vector to each respective SV. After initialization, the BESTPOSB data log for the lead aircraft is no longer required by the instrumentation package (unless a reset is required). However, the data logs for both lead and wing aircraft continue at a 1 Hertz rate. This is because the receiver computed position fix for each aircraft, although a non-DGPS solution, can be utilized to calculate an approximate relative position which is stored in an output file for post-flight data analysis. The impact of transmitting this data log for the lead aircraft over the Freewave serial data link results in less than a 1 Hertz reduction in maximum throughput of the RANGECPMB data log which is discussed next.

The NovAtel RANGECPMB data log was selected to provide the raw GPS measurements. They provide dual-frequency pseudorange and carrier-phase measurements, as well as signal lock times, for each SV in view. They are passed to the CPU at a 20 Hertz rate.

The implementation of the front-end parser consists of three threads that run simultaneously. After initializing the COM port associated with the lead aircraft (reference receiver), the first thread decodes the messages described above and passes them to the parsing thread. Similarly, after initializing the COM port associated with the wing aircraft (mobile receiver), the second thread decodes the messages described above and passes them to the parsing thread. The parsing thread pushes the raw GPS measurements received from the previous two threads onto a memory stack.

When the memory stack contains measurements from both receivers having identical times, those measurements are removed from the stack and a comparison is made. The raw measurements from any SV that is common to both receivers are allowed to pass, while raw measurements from any SV that is not common to both receivers are discarded.

The parsing thread creates two additional sub-threads. The first sub-thread, the Kalman filter task, operates at a 1 Hertz rate. After initialization, and whenever the integer carrier-phase ambiguity set must be re-computed, the Kalman filter task passes data to the ambiguity resolution task. The second sub-thread, the high-rate output task, operates at a 20 Hertz rate. Both sub-threads (Kalman filter task and high-rate output task) as well as the ambiguity resolution task will be thoroughly discussed in later sections within this chapter.

If the user selects post-processed mode, the two COM port threads and parsing thread are not needed. Instead, a simple function is called to process the user provided input file (obtained from a previous data collection using the real-time mode) which contains the raw measurements common to both receivers. After creating the two sub-threads mentioned above (Kalman filter task and high-rate output task), this function processes the file line-by-line and passes the data to the appropriate task in the same manner that the parsing thread does when operating in real-time mode. A significant benefit is that the system can operate approximately ten times faster in post-processed mode than in real time. This allows a great deal of data to be quickly processed and was particularly useful during development and testing.

Because the NovAtel GPS cards provide dual frequency measurements, the question arises as to which measurements to use. If one recalls from Chapter 2, we can not only select L1 or L2 measurements, but can utilize linear combinations of the two measurements such as widelane and narrowlane. To facilitate the real-time switching from widelane measurements to narrowlane measurements, the front-end parser passes both L1 and L2 measurements for each GPS receiver to the Kalman filter task

as well as the high-rate output task. This allows each task to form the appropriate measurements via the use of the following equation:

$$\mathbf{m} = a(\mathbf{m}_{L1}) + b(\mathbf{m}_{L2}) \quad (3.18)$$

where

$\mathbf{m} \equiv$  new measurement vector

$a \equiv$  scalar coefficient of L1 measurement

$\mathbf{m}_{L1} \equiv$  L1 measurement vector

$b \equiv$  scalar coefficient of L2 measurement

$\mathbf{m}_{L2} \equiv$  L2 measurement vector

In the case of widelane measurements,  $a = +1$  and  $b = -1$ . For narrowlane measurements,  $a = +1$  and  $b = +1$ .

### ***3.5 Floating Point Kalman Filter***

The Kalman filter task operates at a 1 Hertz rate and serves several purposes. First, depending on the operating mode selected by the user, the filter provides an approximation of: 1) relative position, 2) relative position, and relative velocity, or 3) relative position, relative velocity, and relative acceleration between the lead and wing aircraft. The reason for giving the user the option of selecting various operating modes is to provide some flexibility regarding the required size of the matrices used in the Kalman filter equations. Assuming a full complement of 12 SVs, the state vector covariance matrix ( $\mathbf{P}$ ) will be a 14x14 matrix in position mode, a 17x17 matrix in position-velocity mode, and a 20x20 matrix in position-velocity-acceleration mode. The increase in the sizes of the other matrices utilized in the Kalman filter equations when going from position mode to position-velocity-acceleration mode are similar as well. Second, regardless of the operating mode selected by the user, the filter provides floating point estimates of the carrier-phase ambiguity values, as well as the

associated covariance matrix, to the ambiguity resolution task discussed in the next section. Finally, the filter computes a matrix containing the unit line-of-sight vectors to each of the SVs and passes it to the high-rate output task which is discussed later in the chapter.

For ease of discussion, the three operating modes will be discussed in parallel. After discussing the initialization for each of the matrices involved, the implementation of the governing Kalman filter equations are discussed.

*3.5.1 Initialization of State Vector.* The state vector contains the relative position vector expressed in Earth Centered Earth Fixed (ECEF) coordinates and the floating point double-difference carrier-phase ambiguities. Additionally, depending on the operating mode selected by the user, it may also contain relative velocity and relative acceleration estimates.

In position mode, the state vector is as follows:

$$\mathbf{x_P} = \begin{bmatrix} X & Y & Z & \Delta\nabla N^{1,2} & \dots & \Delta\nabla N^{1,n} \end{bmatrix}^T \quad (3.19)$$

where

$$x_1 \equiv X \equiv \text{ECEF } X \text{ relative position (m)}$$

$$x_2 \equiv Y \equiv \text{ECEF } Y \text{ relative position (m)}$$

$$x_3 \equiv Z \equiv \text{ECEF } Z \text{ relative position (m)}$$

Note that the remaining terms ( $\Delta\nabla N$ ) are the floating point double-difference carrier-phase ambiguity estimates between the base SV and each of the non-base SVs and are expressed in cycles.

In position-velocity mode, the state vector is as follows:

$$\mathbf{x_{PV}} = \begin{bmatrix} X & Y & Z & \dot{X} & \dot{Y} & \dot{Z} & \Delta\nabla N^{1,2} & \dots & \Delta\nabla N^{1,n} \end{bmatrix}^T \quad (3.20)$$

where

$$x_4 \equiv \dot{X} \equiv \text{ECEF X relative velocity (m/s)}$$

$$x_5 \equiv \dot{Y} \equiv \text{ECEF Y relative velocity (m/s)}$$

$$x_6 \equiv \dot{Z} \equiv \text{ECEF Z relative velocity (m/s)}$$

In position-velocity-acceleration mode, the state vector is as follows:

$$\mathbf{x}_{\text{PVA}} = \left[ X \quad Y \quad Z \quad \dot{X} \quad \dot{Y} \quad \dot{Z} \quad \ddot{X} \quad \ddot{Y} \quad \ddot{Z} \quad \Delta\nabla N^{1,2} \quad \dots \quad \Delta\nabla N^{1,n} \right]^T \quad (3.21)$$

where

$$x_7 \equiv \ddot{X} \equiv \text{ECEF X relative acceleration (m/s}^2\text{)}$$

$$x_8 \equiv \ddot{Y} \equiv \text{ECEF Y relative acceleration (m/s}^2\text{)}$$

$$x_9 \equiv \ddot{Z} \equiv \text{ECEF Z relative acceleration (m/s}^2\text{)}$$

Regardless of the operating mode selected by the user, the first three rows of the state vector are initialized as follows:

$$x_1 = X_{mob} - X_{ref} \quad (3.22)$$

where

$$X_{mob} \equiv \text{ECEF X coordinate of wing aircraft (NovAtel computed)}$$

$$X_{ref} \equiv \text{ECEF X coordinate of lead aircraft (NovAtel computed)}$$

$$x_2 = Y_{mob} - Y_{ref} \quad (3.23)$$

where

$$Y_{mob} \equiv \text{ECEF Y coordinate of wing aircraft (NovAtel computed)}$$

$Y_{ref} \equiv$  ECEF Y coordinate of lead aircraft (NovAtel computed)

$$x_3 = Z_{mob} - Z_{ref} \quad (3.24)$$

where

$Z_{mob} \equiv$  ECEF Z coordinate of wing aircraft (NovAtel computed)

$Z_{ref} \equiv$  ECEF Z coordinate of lead aircraft (NovAtel computed)

The relative velocity and relative acceleration, if selected by the user, are initialized to zero and are updated by the filter.

The double-difference carrier-phase ambiguity terms in the state vector are initialized as follows:

$$\Delta \nabla N^{j,k} = ((\phi_{mob}^j - \phi_{ref}^j)) - (\phi_{mob}^k - \phi_{ref}^k) - (1/\lambda)((\rho_{mob}^j - \rho_{ref}^j) - (\rho_{mob}^k - \rho_{ref}^k)) \quad (3.25)$$

where

j is the base SV

k is the non-base SV

$\lambda$  is the wavelength of the carrier-phase signal

*3.5.2 Initialization of State Vector Covariance Matrix.* The state vector covariance matrix represents the uncertainty of the estimated state vector. The diagonal terms represent the variances and the off-diagonal terms represent the covariances.



In position mode, the state vector covariance matrix is initialized as follows:

$$\mathbf{P_P} = \begin{bmatrix} \sigma_{\Delta X}^2 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_{\Delta Y}^2 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_{\Delta Z}^2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \sigma_{\Delta \nabla N^{1,2}}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta \nabla N^{1,n}}^2 \end{bmatrix} \quad (3.26)$$

where

$\sigma_{\Delta X}^2$  is the variance of relative position (ECEF X coordinate)

$\sigma_{\Delta Y}^2$  is the variance of relative position (ECEF Y coordinate)

$\sigma_{\Delta Z}^2$  is the variance of relative position (ECEF Z coordinate)

$\sigma_{\Delta \nabla N^{j,k}}^2$  is the variance of the double-difference carrier-phase measurement.

In position-velocity mode, the state vector covariance matrix is initialized as follows:

$$\mathbf{P_{PV}} = \begin{bmatrix} \sigma_{\Delta X}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_{\Delta Y}^2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_{\Delta Z}^2 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \sigma_{\Delta \dot{X}}^2 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\Delta \dot{Y}}^2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta \dot{Z}}^2 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta \nabla N^{1,2}}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta \nabla N^{1,n}}^2 \end{bmatrix} \quad (3.27)$$

where

$\sigma_{\Delta \dot{X}}^2$  is the variance of relative velocity (ECEF X coordinate)

$\sigma_{\Delta\dot{Y}}^2$  is the variance of relative velocity (ECEF Y coordinate)

$\sigma_{\Delta\dot{Z}}^2$  is the variance of relative velocity (ECEF Z coordinate)

In position-velocity-acceleration mode, the state vector covariance matrix is initialized as follows:

$$\mathbf{P}_{\text{PVA}} = \begin{bmatrix} \sigma_{\Delta X}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & \sigma_{\Delta Y}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \sigma_{\Delta Z}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \sigma_{\Delta\ddot{X}}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\Delta\ddot{Y}}^2 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta\ddot{Z}}^2 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta\ddot{X}}^2 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta\ddot{Y}}^2 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta\ddot{Z}}^2 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta\nabla N^{1,2}}^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta\nabla N^{1,n}}^2 \end{bmatrix} \quad (3.28)$$

where

$\sigma_{\Delta\ddot{X}}^2$  is the variance of relative acceleration (ECEF X coordinate)

$\sigma_{\Delta\ddot{Y}}^2$  is the variance of relative acceleration (ECEF Y coordinate)

$\sigma_{\Delta\ddot{Z}}^2$  is the variance of relative acceleration (ECEF Z coordinate)

Of note, only the diagonal terms (variances) are set during initialization. The off-diagonal terms (covariances) are populated and updated by the Kalman filter equations discussed later in the chapter.

The initial values of the state vector covariance matrix can be easily changed by the user through modification of a parameter input file which is in an ASCII format.

An in-depth analysis of Kalman filter tuning is provided in Chapter 4. However values that resulted in good filter performance under a variety of conditions are provided in Table 3.1 below:

Table 3.1: Initial Covariance Values

relative position	$\sigma_p^2 = (5m)^2$
relative velocity	$\sigma_v^2 = (5m/s)^2$
relative acceleration	$\sigma_a^2 = (5m/s^2)^2$
$\Delta \nabla N$	$\sigma_N^2 = (\frac{50}{\lambda} cycles)^2$

*3.5.3 Initialization of Transition Matrix.* From Chapter 2, the fundamental or transition matrix,  $\Phi$ , was shown to be:

$$\Phi(\mathbf{T}_s) = \mathbf{e}^{\mathbf{F}\mathbf{T}_s} \quad (3.29)$$

where

$\mathbf{T}_s$  is the measurement sampling period.

However, instead of numerically calculating a matrix exponential, the system presented in this thesis constructs the  $\Phi$  matrix as discussed below.

In position-velocity-acceleration mode, the  $\Phi$  matrix is constructed as follows [17]:

$$\Phi_{PVA}(T_s) = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & A & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & A & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & A & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & B & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & B & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & B & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & C & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & C & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.30)$$

where

$\Delta t$  is the sampling rate of the Kalman filter (1 second)

$$A = T_s^2(e^{-\Delta t/T_s} - 1) + T_s\Delta t$$

$$B = T_s(1 - e^{-\Delta t/T_s})$$

$$C = e^{-\Delta t/T_s}$$

In position-velocity mode, the first three rows and the first three columns of Equation (3.30) are removed to yield the following [17]:

$$\Phi_{PV}(T_s) = \begin{bmatrix} 1 & 0 & 0 & B & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & B & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & B & 0 & \dots & 0 \\ 0 & 0 & 0 & C & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & C & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & C & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.31)$$

In position mode, the  $\Phi$  matrix is simply the identity matrix:

$$\Phi_P(T_s) = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.32)$$

*3.5.4 Initialization of Discrete Noise Matrix.* In position-velocity-acceleration mode, the process noise is modeled as a First-Order Gauss-Markov Approximation

(FOGMA) and the discrete noise matrix,  $\mathbf{Qd}$ , is computed as follows [17]:

$$\mathbf{Qd}_{\text{PVA}} = \begin{bmatrix} D & 0 & 0 & E & 0 & 0 & G & 0 & 0 & 0 & \dots & 0 \\ 0 & D & 0 & 0 & E & 0 & 0 & G & 0 & 0 & \dots & 0 \\ 0 & 0 & D & 0 & 0 & E & 0 & 0 & G & 0 & \dots & 0 \\ E & 0 & 0 & K & 0 & 0 & L & 0 & 0 & 0 & \dots & 0 \\ 0 & E & 0 & 0 & K & 0 & 0 & L & 0 & 0 & \dots & 0 \\ 0 & 0 & E & 0 & 0 & K & 0 & 0 & L & 0 & \dots & 0 \\ G & 0 & 0 & L & 0 & 0 & M & 0 & 0 & 0 & \dots & 0 \\ 0 & G & 0 & 0 & L & 0 & 0 & M & 0 & 0 & \dots & 0 \\ 0 & 0 & G & 0 & 0 & L & 0 & 0 & M & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & U & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & U \end{bmatrix} \quad (3.33)$$

where

$$D \equiv \frac{1}{2}T_s^5 q_v a (1 - e^{-2\Delta t/T_s}) + T_s^4 q_v a \Delta t (1 - 2e^{-\Delta t/T_s}) - T_s^3 q_v a (\Delta t)^2 + \frac{1}{3}T_s^2 q_v a (\Delta t)^3$$

$$E \equiv T_s^4 q_v a (\frac{1}{2}e^{-2\Delta t/T_s} - e^{-\Delta t/T_s} + \frac{1}{2}) + T_s^3 q_v a \Delta t (e^{-\Delta t/T_s} - 1) + \frac{1}{2}T_s^2 q_v a (\Delta t)^2$$

$$G \equiv \frac{1}{2}T_s^3 q_v a (1 - e^{-2\Delta t/T_s}) - T_s^2 q_v a \Delta t e^{-\Delta t/T_s}$$

$$K \equiv \frac{1}{2}T_s^3 q_v a (-e^{-2\Delta t/T_s} + 4e^{-\Delta t/T_s} + 2\frac{\Delta t}{T_s} - 3)$$

$$L \equiv -\frac{1}{2}T_s^2 q_v a (-e^{-2\Delta t/T_s} + 2e^{-\Delta t/T_s} - 1)$$

$$M \equiv -\frac{1}{2}T_s q_v a (e^{-2\Delta t/T_s} - 1)$$

$$U \equiv q_N \Delta t$$

In position-velocity mode, the process noise is also modeled using a FOGMA. As was the case for the  $\Phi$  matrix, the first three rows and the first three columns of

Equation (3.33) are removed to yield the following:

$$\mathbf{Qd}_{\mathbf{PV}} = \begin{bmatrix} K & 0 & 0 & L & 0 & 0 & 0 & \dots & 0 \\ 0 & K & 0 & 0 & L & 0 & 0 & \dots & 0 \\ 0 & 0 & K & 0 & 0 & L & 0 & \dots & 0 \\ L & 0 & 0 & M & 0 & 0 & 0 & \dots & 0 \\ 0 & L & 0 & 0 & M & 0 & 0 & \dots & 0 \\ 0 & 0 & L & 0 & 0 & M & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & U & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & U \end{bmatrix} \quad (3.34)$$

In position mode, the process noise is modeled using a random walk and  $\mathbf{Qd}$  matrix is simply:

$$\mathbf{Qd}_{\mathbf{PV}} = \begin{bmatrix} q_p \Delta t & 0 & 0 & 0 & \dots & 0 \\ 0 & q_p \Delta t & 0 & 0 & \dots & 0 \\ 0 & 0 & q_p \Delta t & 0 & \dots & 0 \\ 0 & 0 & 0 & q_N \Delta t & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & q_N \Delta t \end{bmatrix} \quad (3.35)$$

The noise characteristics of the Kalman filter can be easily modified by the user through the parameter input file. As mentioned during the discussion of the initialization of the state vector covariance matrix, aspects relating to filter tuning are deferred until Chapter 4. However, values that were determined to result in good filter performance under a variety of conditions are shown in Table 3.2:

*3.5.5 Initialization of Measurement Covariance Matrix.* The measurement covariance matrix represents the uncertainty of the measurements and is not dependent on the operating mode selected by the user. Instead, it is only dependent on the

Table 3.2: Initial Noise Values

parameter	position	position-velocity	position-velocity-acceleration
$T_s$	3 sec	3 sec	3 sec
$q_{pva}$	$q_p = 15$	$q_{av} = \frac{2\sigma_v^2}{T_s}$	$q_{av} = \frac{2\sigma_a^2}{T_s}$
$q_N$	1.1e-2	1.1e-2	1.1e-2

number of visible SVs which are common to both receivers. However, there are five cases which must be considered.

The first case is determination of the code variances. Recall from Equation (2.1):

$$\rho = r + c(\delta t_r - \delta t_{sv}) + T + I + m_\rho + v_\rho \quad (3.36)$$

Because the double differencing of the measurements eliminated both receiver and satellite clock error, the second and third terms on the right hand side of the equation may be discounted. Because the application presented in this thesis involves UAV's in formation flight, i.e. very short baselines, the impact of tropospheric and ionospheric errors will be ignored. Hence, the only remaining source of error is multipath and noise. The code variance will therefore be a conservative best guess of the average deviation due to these two error sources.

The second case is the determination of the phase variances. Recall from Equation (2.2):

$$\phi = \lambda^{-1}(r + c(\delta t_r - \delta t_{sv}) + T - I + m_\phi + v_\phi) + N \quad (3.37)$$

Using the same methodology as for the code variance, the phase variance is driven only by multipath and noise errors and will differ from code variance only in magnitude and units.

The third case is the determination of the code covariances. If one assumes that each visible SV has a relatively unique bearing and elevation, then the impact of multipath and noise errors will be different for each measurement and the covariances (off-diagonal terms) would be zero. However, the use of double-difference measure-



ments with a common SV complicates things slightly. Because one of the two SVs in each double-difference measurements is identical for each measurement, the code covariances will be exactly half the value of the code variances.

The fourth case is the determination of the phase covariances. Using the same methodology as for code covariances, it can easily be seen that the phase covariance values must be exactly half the value of the phase variances.

The fifth and final case is the determination of the code/phase cross-covariances. Unless the baseline between GPS receivers grows large enough for tropospheric errors to become significant, the code and phase measurement errors are uncorrelated and the code/phase cross-covariances are zero.

By combining the five aforementioned cases, the complete measurement covariance matrix with five visible SVs is as follows:

$$\mathbf{R} = \begin{bmatrix} a & c & c & c & 0 & 0 & 0 & 0 \\ c & a & c & c & 0 & 0 & 0 & 0 \\ c & c & a & c & 0 & 0 & 0 & 0 \\ c & c & c & a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b & d & d & d \\ 0 & 0 & 0 & 0 & d & b & d & d \\ 0 & 0 & 0 & 0 & d & d & b & d \\ 0 & 0 & 0 & 0 & d & d & d & b \end{bmatrix} \quad (3.38)$$

where

$a \equiv$  code variances (Case I)

$b \equiv$  phase variances (Case II)

$c \equiv$  code covariances (Case III)

$d \equiv$  phase covariances (Case IV)

Values which were experimentally determined [18] are shown in Table 3.3 below:

Table 3.3: Initial Values for Measurement Covariance Matrix

parameter	value
code variance	10.24
phase variance	0.0994
code covariance	5.12
phase covariance	0.0497

*3.5.6 Implementation of Real-Time Considerations.* After the state vector, state vector covariance matrix, transition matrix, discrete noise matrix, and measurement covariance matrix have been initialized, the Kalman filter is ready to begin processing measurements. However, after each new measurement is received, several tests must be performed to ensure that the added complexity due to real-time considerations does not cause the filter to become unstable.

The first test performed by the filter determines whether or not measurements exist for the base SV. If the base SV is still in view, the algorithm proceeds to the second test. If, however, the base SV is no longer available, then a new base SV is selected and both the state vector and the state vector covariance matrix are transformed utilizing Equation (3.7) and Equation (3.8).

The second test performed by the filter determines whether or not any SVs have been gained or lost since the previous epoch. For each new SV, a row is added to the state vector and both a row and a column are added to the state vector covariance matrix. For each SV that is no longer in view, a row is deleted from the state vector and both a row and a column are deleted from the state vector covariance matrix. Of note, should the base SV be lost, the first test will have already performed the necessary transformation of variables to switch the base SV.

The third test performed by the filter determines whether or not the total number of visible SVs has changed. If there is no change, the filter proceeds to the fourth and final test. If the number of visible SVs has changed, then the measurement covariance matrix ( $\mathbf{R}$ ), transition matrix ( $\Phi$ ), and discrete noise matrix ( $\mathbf{Qd}$ ) are resized

to the proper dimensions. The reason for conducting this test separately from the previous two tests is as follows. During testing, there were several instances of a new SV coming into view and an existing SV becoming unavailable during the 1 second time interval between Kalman filter updates. Should this happen, there is no need to resize the above mentioned matrices since the total number of SVs is the same.

The fourth and final test performed by the filter determines whether or not there has been a cycle slip. Recall from Chapter 3 that a cycle slip is a momentary lapse of SV availability. The significance of a cycle slip becomes apparent when one considers that the GPS receiver must start the accumulation of Doppler over again. This results in a new carrier-phase measurement and a new carrier-phase ambiguity value. The NovAtel RANGECPB data log provides a signal lock time for both L1 and L2 measurements. This signal lock time is utilized by the filter for cycle slip detection. In the event of a cycle slip, the affected double-difference carrier-phase measurement in the state vector is re-initialized using Equation (3.25). Additionally, the affected row and column of the state vector covariance matrix are re-initialized with the covariances set to zero and the variance set to the value stored in the user defined parameter input file.

### *3.5.7 Time Propagation of State Vector and State Vector Covariance Matrix.*

After the filter ensures that all real time considerations have been properly dealt with, it is ready to propagate the state vector and state vector covariance matrix forward in time using the equations shown in Section 2.4.2.

*3.5.8 Formulation of Measurement Equation.* In Chapter 2 it was shown that the measurement vector must be related to the state vector. For convenience, the relationship, provided in Equation (2.18), is repeated below:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (3.39)$$

The Kalman filter presented in this thesis utilizes double-difference pseudorange and double-difference carrier-phase measurements. For ease of explanation, it is assumed that there are only 4 SVs with SV 1 selected as the base SV. However, the concepts shown here can be easily extended to other cases. With the preceding assumptions, the left side of Equation (3.39) is defined as follows:

$$\mathbf{z} = \begin{bmatrix} \Delta\nabla\rho_{A,B}^{1,2} \\ \Delta\nabla\rho_{A,B}^{1,3} \\ \Delta\nabla\rho_{A,B}^{1,4} \\ \Delta\nabla\phi_{A,B}^{1,2} \\ \Delta\nabla\phi_{A,B}^{1,3} \\ \Delta\nabla\phi_{A,B}^{1,4} \end{bmatrix} \quad (3.40)$$

where

A is the mobile receiver (wing aircraft)

B is the reference receiver (lead aircraft)

The next step is to determine a measurement matrix ( $\mathbf{H}$ ) that when post-multiplied by the state vector ( $\mathbf{x}$ ) is, with the exception of the zero-mean white Gaussian noise, equal to the measurement vector ( $\mathbf{z}$ ). To facilitate the discussion, refer to Figure 3.2.

Because the distance between the SV and each GPS receiver, approximately 20 million meters, is much greater than the distance between the two aircraft, it can be assumed that for any given SV, both aircraft share the same unit vector line of sight to the SV. In Figure 3.2, this unit line of sight vector (for the  $j^{th}$  SV) is annotated as  $\mathbf{e}^j$ . Additionally, an inspection of the geometry present in Figure 3.2 reveals:

$$r_{mob}^j - r_{ref}^j = \Delta\mathbf{x} \cdot \mathbf{e}^j \quad (3.41)$$

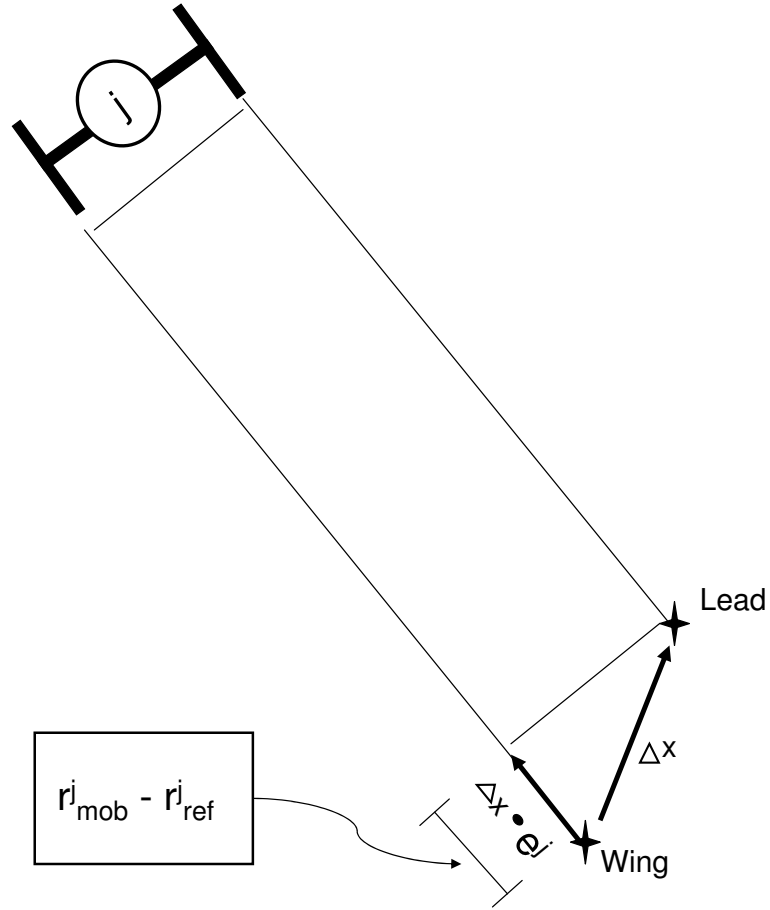


Figure 3.2: Formulation of H matrix

Hence, the difference in ranges between the  $j^{th}$  SV and each receiver is related to the state vector and the unit line of sight vector to the  $j^{th}$  SV through the use of Equation (3.41).

Remembering that the double-difference pseudorange measurement is defined as:

$$\Delta \nabla \rho_{AB}^{jk} = \Delta \rho_{AB}^j - \Delta \rho_{AB}^k \quad (3.42)$$

and the pseudorange is defined as:

$$\rho = r + c(\delta t_r - \delta t_{sv}) + T + I + m_\rho + v_\rho \quad (3.43)$$

One can utilize Equation (3.43) to expand Equation (3.42), neglect insignificant (or zero) terms, and obtain the following:

$$\Delta\nabla\rho_{AB}^{jk} = (r_A^j - r_B^j) - (r_A^k - r_B^k) \quad (3.44)$$

Utilizing Equation (3.41), Equation (3.44) can be re-written as:

$$\Delta\nabla\rho_{AB}^{jk} = (\Delta\mathbf{x} \cdot \mathbf{e}^j) - (\Delta\mathbf{x} \cdot \mathbf{e}^k) \quad (3.45)$$

Rearranging terms yields:

$$\Delta\nabla\rho_{AB}^{jk} = (\mathbf{e}^j - \mathbf{e}^k) \cdot \Delta\mathbf{x} \quad (3.46)$$

A similar analysis for the case of double-difference carrier-phase measurements yields the following:

$$\Delta\nabla\phi_{AB}^{jk} = \lambda^{-1}(\mathbf{e}^j - \mathbf{e}^k) \cdot \Delta\mathbf{x} + \Delta\nabla N_{AB}^{jk} \quad (3.47)$$

Using Equation (3.46) and Equation (3.47) to build the complete H matrix, Equation (3.39), neglecting noise, becomes:

$$\begin{bmatrix} \Delta\nabla\rho_{A,B}^{1,2} \\ \Delta\nabla\rho_{A,B}^{1,3} \\ \Delta\nabla\rho_{A,B}^{1,4} \\ \Delta\nabla\phi_{A,B}^{1,2} \\ \Delta\nabla\phi_{A,B}^{1,3} \\ \Delta\nabla\phi_{A,B}^{1,4} \end{bmatrix} = \begin{bmatrix} (\mathbf{e}^1 - \mathbf{e}^2) & 0 & 0 & 0 \\ (\mathbf{e}^1 - \mathbf{e}^3) & 0 & 0 & 0 \\ (\mathbf{e}^1 - \mathbf{e}^4) & 0 & 0 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^2) & 1 & 0 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^3) & 0 & 1 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^4) & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\nabla N_{A,B}^{1,2} \\ \Delta\nabla N_{A,B}^{1,3} \\ \Delta\nabla N_{A,B}^{1,4} \end{bmatrix} \quad (3.48)$$

With the above mentioned matrices now computed, the state vector ( $\mathbf{x}$ ) and state covariance matrix ( $\mathbf{P}$ ) can be updated utilizing the equations in Section 2.4.2.

This concludes the section on the Kalman filter. In review, the filter operates at a 1 Hertz rate and receives raw GPS measurements as input. It provides three separate

outputs. First, it provides the floating point estimate of the carrier-phase ambiguities, and the associated covariance matrix, to the ambiguity resolution routine. Second, it provides a matrix of unit line-of-sight vectors ( $\mathbf{e}$ ) from the wing aircraft to each of the SVs. This is used by the high-rate output routine to calculate the measurement matrix ( $\mathbf{H}$ ). The reason for passing the  $\mathbf{e}$  matrix instead of the  $\mathbf{H}$  matrix is that the  $\mathbf{H}$  matrix for the Kalman filter contains both code and phase data. In contrast, the  $\mathbf{H}$  matrix utilized by the high-rate output task contains only phase data. Finally, the Kalman filter provides an independent approximate relative position output, as well as relative velocity and relative acceleration, if selected by the user.

### ***3.6 Carrier-Phase Ambiguity Resolution***

The carrier-phase ambiguity task is summoned by the Kalman filter task whenever the integer ambiguity set must be determined from the state vector and covariance matrix. After the floating point estimates of the carrier-phase ambiguities, along with the associated covariance matrix, are received from the Kalman filter, the ambiguity resolution task passes these matrices to a LAMBDA routine which returns two possible integer ambiguity sets and an associated value, referred to as the squared normal, for each set. The squared normal value represents how good the LAMBDA routine believes the fit is and is nothing more than the distance between the respective integer ambiguity candidate set and the floating point ambiguity set. Of note, the ambiguity resolution routine does not have to operate continuously. Instead, once the ambiguity estimates have been correctly determined, the high-rate task uses the known integer ambiguities to determine the integer ambiguities for any newly acquired SVs. This aspect will be discussed in greater detail in the next section.

The LAMBDA routine, coded in C++ by the author, is nearly identical to the lambda2.m routine which is publicly available through Delft University [5]. The function calls each of the subroutines used by the lambda2.m routine are also nearly identical. As Teunissen, Jonge, and Tiberius [5] provide an excellent description of

the MATLAB implementation, a detailed discussion here is not warranted. However, there are several items that do require further discussion.

First, the floating point values and covariance matrix passed to the ambiguity routine are only those associated with the carrier-phase ambiguities. The Kalman filter data for relative position, relative velocity, and relative acceleration is not needed and is stripped off before passing the matrices to the ambiguity resolution task.

Second, as mentioned above, the LAMBDA routine returns two possible integer ambiguity sets. This is because the ambiguity set with the lowest squared normal value (i.e., the best fit) may not in fact be the correct set. Although the correct ambiguity set generally has the lowest value, there will occasionally be times where an incorrect ambiguity set has a value that briefly dips below that of the correct set before rising back to an elevated level. An example of this is given in Section 4.4.2. However, a means is required to enable the selection of the correct ambiguity despite this phenomena.

As discussed by Teunissen [20], one of the earliest and most popular ways of validating the integer ambiguity solution is through the use of a ratio test. As the name implies, a value is formed that is the ratio of the squared normal value of the “second-best” ambiguity set to the squared normal value of the “best” ambiguity set. When this ratio is below a certain user defined threshold, referred to as the critical value, the computed integer ambiguity set is not known with confidence. Typically, the critical value is above 1.5 and below 5.0, however it may be as high as 10 [20]. In cases where the ratio is below the critical value, the user must discard the integer solution and can either proceed by using the floating point solution or wait for the two squared normal values to diverge which will result in a ratio that is above the critical value.

The instrumentation package presented in this thesis utilizes the ratio test discussed above. If the ratio is greater than 2.5, the algorithm assumes that the ambiguity set associated with the low squared normal value is correct and passes the ambiguity



set to the high-rate output task. However, if the ratio is less than this, the ambiguity resolution routine waits for the Kalman filter to provide an updated state vector and covariance matrix. After passing these updated matrices to the LAMBDA routine and receiving updated ambiguity sets and squared normal values, the ratio test is repeated. Once the ratio test passes, the ambiguity resolution task returns a flag to the Kalman filter task which lets the filter know that the carrier-phase ambiguities have been resolved to their integer values. Once this flag is set, the Kalman filter no longer passes the state vector and covariance matrix to the ambiguity resolution task unless a reset is required. Results for the length of time to pass the ratio test and the success rate of selecting the proper ambiguity set are presented in Chapter 4.

### ***3.7 High-Rate Relative Position Output***

The high-rate output task operates at a 20 Hertz rate. Because of the desire to obtain an extremely precise solution while simultaneously minimizing system latency, a key objective was to make this task as efficient as possible. As a result, the only raw measurements utilized by the high-rate task are the double-difference carrier-phase measurements. By using only the phase measurements, and discarding the code measurements, the size of the matrices involved are significantly reduced.

*3.7.1 Formulation of Relative Position Vector.* For ease of discussion, as was done in the discussion on Kalman filtering, it will be assumed that there are four SVs with the base SV being PRN 1. From Chapter 2, recall that the measurements are related to the state vector as follows:

$$\begin{bmatrix} \Delta \nabla \rho_{A,B}^{1,2} \\ \Delta \nabla \rho_{A,B}^{1,3} \\ \Delta \nabla \rho_{A,B}^{1,4} \\ \Delta \nabla \phi_{A,B}^{1,2} \\ \Delta \nabla \phi_{A,B}^{1,3} \\ \Delta \nabla \phi_{A,B}^{1,4} \end{bmatrix} = \begin{bmatrix} (\mathbf{e}^1 - \mathbf{e}^2) & 0 & 0 & 0 \\ (\mathbf{e}^1 - \mathbf{e}^3) & 0 & 0 & 0 \\ (\mathbf{e}^1 - \mathbf{e}^4) & 0 & 0 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^2) & 1 & 0 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^3) & 0 & 1 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^4) & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \nabla N_{A,B}^{1,2} \\ \Delta \nabla N_{A,B}^{1,3} \\ \Delta \nabla N_{A,B}^{1,4} \end{bmatrix} \quad (3.49)$$

After eliminating the double-difference code measurements, Equation (3.49) becomes:

$$\begin{bmatrix} \Delta \nabla \phi_{A,B}^{1,2} \\ \Delta \nabla \phi_{A,B}^{1,3} \\ \Delta \nabla \phi_{A,B}^{1,4} \end{bmatrix} = \begin{bmatrix} \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^2) & 1 & 0 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^3) & 0 & 1 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^4) & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \nabla N_{A,B}^{1,2} \\ \Delta \nabla N_{A,B}^{1,3} \\ \Delta \nabla N_{A,B}^{1,4} \end{bmatrix} \quad (3.50)$$

It is desired to separate the components of the relative position vector from the other terms in Equation (3.50). In order to do this, the relative position terms are separated from the double-difference carrier-phase ambiguities as follows:

$$\begin{bmatrix} \Delta \nabla \phi_{A,B}^{1,2} \\ \Delta \nabla \phi_{A,B}^{1,3} \\ \Delta \nabla \phi_{A,B}^{1,4} \end{bmatrix} = \begin{bmatrix} \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^2) \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^3) \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^4) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \nabla N_{A,B}^{1,2} \\ \Delta \nabla N_{A,B}^{1,3} \\ \Delta \nabla N_{A,B}^{1,4} \end{bmatrix} \quad (3.51)$$

Rearranging terms in Equation (3.51) yields the following:

$$\begin{bmatrix} \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^2) \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^3) \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^4) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} \Delta \nabla \phi_{A,B}^{1,2} \\ \Delta \nabla \phi_{A,B}^{1,3} \\ \Delta \nabla \phi_{A,B}^{1,4} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \nabla N_{A,B}^{1,2} \\ \Delta \nabla N_{A,B}^{1,3} \\ \Delta \nabla N_{A,B}^{1,4} \end{bmatrix} \quad (3.52)$$

In general form, Equation (3.52) can be expressed as follows:

$$\mathbf{H}\mathbf{x} = \mathbf{z} \quad (3.53)$$

Because the  $\mathbf{H}$  matrix is not a square matrix, the relative position can not be isolated by simply multiplying by  $\mathbf{H}^{-1}$ . Instead, a pseudoinverse must be used as follows:

$$\mathbf{x} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{z} \quad (3.54)$$

where:

$$\mathbf{x} \equiv \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \quad (3.55)$$

$$\mathbf{H} \equiv \begin{bmatrix} \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^2) \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^3) \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^4) \end{bmatrix} \quad (3.56)$$

$$\mathbf{z} \equiv \begin{bmatrix} \Delta \nabla \phi_{A,B}^{1,2} \\ \Delta \nabla \phi_{A,B}^{1,3} \\ \Delta \nabla \phi_{A,B}^{1,4} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \nabla N_{A,B}^{1,2} \\ \Delta \nabla N_{A,B}^{1,3} \\ \Delta \nabla N_{A,B}^{1,4} \end{bmatrix} \quad (3.57)$$

It can be seen that solving the right side of Equation (3.54) yields the relative position vector, which is the ultimate goal.

Recall from the discussion of the Kalman filter task that the  $\mathbf{H}$  matrix was defined as:

$$\mathbf{H} = \begin{bmatrix} (\mathbf{e}^1 - \mathbf{e}^2) & 0 & 0 & 0 \\ (\mathbf{e}^1 - \mathbf{e}^3) & 0 & 0 & 0 \\ (\mathbf{e}^1 - \mathbf{e}^4) & 0 & 0 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^2) & 1 & 0 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^3) & 0 & 1 & 0 \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^4) & 0 & 0 & 1 \end{bmatrix} \quad (3.58)$$

By inspection, one can see that the  $\mathbf{H}$  matrix used in Equation (3.56) is nothing more than the lower left quadrant of Equation (3.58). The benefit of using the reduced  $\mathbf{H}$  matrix becomes apparent when one realizes that the term  $(\mathbf{H}^T \mathbf{H})$  is always a 3x3 matrix. Thus, the inverse which must be computed every 0.05 seconds by the high-rate task is always a 3x3 matrix and requires far less computational power than that required by the Kalman filter which must compute the inverse of the  $\mathbf{H}$  matrix where the dimensions can be as high as 22x22.

There are two components in Equation (3.57) which must now be discussed. The first are the double-difference carrier-phase measurements. These are obtained directly from the front end parser at a 20 Hertz rate. The second are the double-difference carrier-phase ambiguities, resolved to their integer values. These are initially obtained from the ambiguity resolution task. However, when a new SV comes into view, it is not necessary to recompute the entire ambiguity set from scratch. Instead, the double-difference carrier-phase integer ambiguity associated with the newly acquired SV can be determined as discussed below.

Equation (3.51) can be rearranged as follows:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \nabla N_{A,B}^{1,2} \\ \Delta \nabla N_{A,B}^{1,3} \\ \Delta \nabla N_{A,B}^{1,4} \end{bmatrix} = \begin{bmatrix} \Delta \nabla \phi_{A,B}^{1,2} \\ \Delta \nabla \phi_{A,B}^{1,3} \\ \Delta \nabla \phi_{A,B}^{1,4} \end{bmatrix} - \begin{bmatrix} \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^2) \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^3) \\ \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^4) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \quad (3.59)$$

Now consider the case where a new SV, PRN 5, comes into view. Inspection of Equation (3.59) reveals that the new double-difference carrier-phase measurement is as follows:

$$\Delta\nabla N^{1,5} = \Delta\nabla\phi_{A,B}^{1,5} - \lambda^{-1}(\mathbf{e}^1 - \mathbf{e}^5)\mathbf{x} \quad (3.60)$$

Thus, in order to compute the double-difference carrier-phase ambiguity in Equation (3.60), the only unknown value we need is the relative position vector. Initially it may appear that we need the double-difference carrier-phase ambiguity term to calculate the relative position vector, and vice-versa. However, by initially ignoring the newly acquired SV, and calculating the relative position without it, formation of the  $\mathbf{x}$  vector is possible. This relative position vector ( $\mathbf{x}$ ) can now be used in Equation (3.60) to calculate the ambiguity value. By simple rounding of the result, the ambiguity value can be resolved to its integer value and saved for use during future epochs.

The preceding discussion has shown that in order to calculate the high-rate relative position, there are three required pieces of information. First, the reduced  $\mathbf{H}$  matrix must be derived from the  $\mathbf{e}$  matrix which is obtained from the Kalman filter task operating at a 1 Hertz rate. Second, the double-difference carrier-phase measurements must be obtained from the front end parser at a 20 Hertz rate. Finally, the integer values of the carrier-phase ambiguities must be obtained. Initially the integer ambiguity set is obtained by the ambiguity resolution task. However, after the ambiguity set has been determined, the integer value of the double-difference carrier-phase ambiguity for any newly acquired SV is determined internally by the high-rate output task as previously discussed. Thus, by combining these three pieces of data, through the use of Equation (3.54), the high-rate relative position vector may be computed.

*3.7.2 Additional Real-Time Complexity.* As was the case for the Kalman filter task, the requirement that the system operate in a RTK environment creates added complexity. For the high-rate output task, this complexity is relatively specific and is due to the latency associated with updating the  $\mathbf{H}$  matrix at a reduced 1 Hertz

rate. Additionally, the integer ambiguity set must be adjusted for the gain and/or loss of SVs. There are three instances, discussed below, which must be considered.

*3.7.2.1 Gain of SV.* If a new SV comes into view after the unit line-of-sight vector ( $\mathbf{e}$ ) has been passed by the Kalman filter, the process is relatively straight forward. Because it is impossible to form an  $\mathbf{H}$  matrix containing a unit line-of-sight vector that has yet to be calculated, the newly acquired SV is simply ignored for up to 0.95 seconds until it is seen by the Kalman filter task and a unit line-of-sight vector computed. In terms of carrier-phase ambiguities, the newly acquired SV is excluded from use until the  $\mathbf{e}$  matrix which includes the new SV is formulated by the Kalman filter and passed to the high-rate task. At this point, the precise relative position vector is computed, again ignoring the new SV. Finally, after the precise relative position is known, it is used to calculate the new ambiguity value. This value is then saved for use during all subsequent epochs.

*3.7.2.2 Loss of non-base SV.* If an existing SV becomes unavailable after the unit line-of-sight vector ( $\mathbf{e}$ ) is calculated, the process is also relatively straight forward. When formulating the  $\mathbf{H}$  matrix, the unit line-of-sight for the lost SV is simply ignored. In terms of carrier-phase ambiguities, the double-difference carrier-phase ambiguity associated with the lost SV is no longer of any use and is therefore eliminated.

*3.7.2.3 Loss of base SV.* The most complicated real-time occurrence which must be considered is the loss of the base SV after the ( $\mathbf{e}$ ) matrix is computed. In this case, a transformation similar to the transformation done by the Kalman filter must be performed. As was the case for the Kalman filter task, after the transformation to a new base SV is complete, any term containing data associated with the old base SV (SV which was lost) must be deleted from further consideration.

### 3.8 *Summary*

This chapter has fully described the development of the algorithm utilized by the instrumentation package. After the raw data is synchronized by a front end parser, there are two separate tasks which operate simultaneously but asynchronously.

The Kalman filter task, operating at 1 Hertz, provides a rough estimate of relative position, and if selected by the user, relative velocity and relative acceleration. Additionally, it provides the unit line-of-sight information required to allow the high-rate output task to construct the  $\mathbf{H}$  matrix. Finally, when required, it provides a floating point estimate of double-difference carrier-phase ambiguities, and the associated covariance matrix, to the ambiguity resolution task. When summoned by the Kalman filter, the ambiguity resolution task utilizes the LAMBDA method, and passes the integer carrier-phase ambiguity values to the high-rate output task.

The high-rate output task receives the previously mentioned data, along with raw GPS measurements at a 20 Hertz rate, and provides as output a low-latency, 20 Hertz, precise relative position which is accurate to a few centimeters.

## IV. Results and Analysis

This chapter provides the results and analysis of the testing which was performed on the instrumentation package presented in this thesis. After describing the methodology used to test performance, a brief discussion is provided on how the truth values were obtained. Next, an example is provided to illustrate how DGPS can obtain higher accuracy than stand-alone GPS. After providing results and analysis for the ground based static tests, the chapter concludes by presenting the results and analysis for the ground based dynamic tests.

### 4.1 *Testing Methodology*

In order to fully determine the effectiveness of the instrumentation package presented in this thesis, one must be concerned with more than just the preciseness of the high-rate output. In fact, each of the three tasks discussed in the previous chapter (Kalman filtering, ambiguity resolution, and high-rate output) must be evaluated.

*4.1.1 Kalman Filter Task.* Because the Kalman filter provides floating point estimates of the carrier-phase ambiguities and an associated covariance matrix to the ambiguity resolution task, it is imperative that the filter be properly tuned. One means of evaluating filter tuning is to compare the error in the state vector to the expected error statistics as represented by the covariance matrix.

Additionally, one of the roles of the Kalman filter is to dampen the noisy nature of the raw GPS measurements. In order to evaluate its effectiveness in accomplishing this task, we can compare the error in the filter calculated relative position to the error in the NovAtel single point calculated relative position.

*4.1.2 Ambiguity Resolution Task.* As was discussed in Chapter 3, the ambiguity set returned from the LAMBDA routine having the lowest squared normal value is not always the correct ambiguity set. To visualize the typical characteristics of the squared normal value, a comparison can be made between the value returned for the correct set and the value returned for an incorrect set.



In order to obtain the desired precision in the high-rate output task, the ambiguity resolution task must resolve the double-difference carrier-phase ambiguities to their integer values. One way of evaluating performance in this area is to conduct a large number of data runs, re-initializing the Kalman filter each time. After the data runs are complete, the performance statistics describing the length of time required to determine the ambiguity set and the percentage of correct ambiguity sets can be calculated.

*4.1.3 high-rate Output Task.* In the case of the high-rate output task, there are two items of concern, precision and latency. In order to evaluate precision, a comparison is made between the calculated relative position and the actual relative position.

To evaluate system latency, the time from the reception of the raw GPS measurements to the time that the use of those measurements resulted in a relative position solution is calculated. Additionally, the interval between solutions is determined to ensure that solutions are being provided at a steady state rate.

## **4.2 Truth Values**

In order to determine errors in relative position, the actual relative position must be known. For the ground based static tests this was quite simple. Raw GPS measurements for both receivers were collected at a 30 second interval for 24 hours. These measurements were then electronically submitted to the Online Positioning User Service (OPUS) [12]. After waiting for precise orbital information to become available, the actual ECEF position for each GPS antenna was returned via e-mail. Taking the difference between these two known absolute positions yields the known relative position. The truth values used during static testing are shown in Table 4.1.

Truth values for dynamic testing are more difficult in nature. One possibility is to compare the change in reported relative position to the change in actual relative position when the antennas are moved from a start point to an end point. The change

Table 4.1: Truth Values Utilized During Ground Static Testing

	ECEF X (m)	ECEF Y (m)	ECEF Z (m)
reference receiver	506,070.120	-4,882,258.343	4,059,613.328
standard deviation	0.019	0.016	0.006
mobile receiver	506,072.907	-4,882,262.297	4,059,609.497
standard deviation	0.027	0.019	0.011
relative position (reference - mobile)	-2.787	3.954	3.831

in reported relative position is obtained directly from the change in the high-rate output. The change in actual relative position is obtained by physically measuring the distance between the two points. However, a better means of obtaining dynamic truth values, which was utilized during the dynamic testing discussed later in the chapter, is to incorporate two reference receivers and two mobile receivers. In this case, both reference receivers are connected to a common antenna through a coaxial cable splitter. Additionally, both mobile receivers are connected to a common antenna through another coaxial cable splitter. One pair of receivers is utilized by the instrumentation package presented in this thesis. The second pair is utilized by an independent, commercially available, DGPS application. Post processing analysis can be performed and a comparison made between the two independent solutions.

### 4.3 The Benefit of DGPS - An Example

Before examining the performance of the DGPS application presented in this thesis, it is beneficial to briefly look at the errors that exist in a typical stand-alone application and how DGPS can improve performance. To illustrate this, consider the accuracy of the stand-alone absolute position for each of the two locations used during static testing. Figure 4.1 depicts the East-North-Up (ENU) errors for each of the two locations during a 60 minute data run.

One can see that in the horizontal plane, the magnitude of the error is generally within 1 meter. In the vertical plane, accuracy is slightly worse and the magnitude of the errors is approximately twice that found for the horizontal plane. These values are consistent with the generally agreed upon precision expected from a dual frequency,

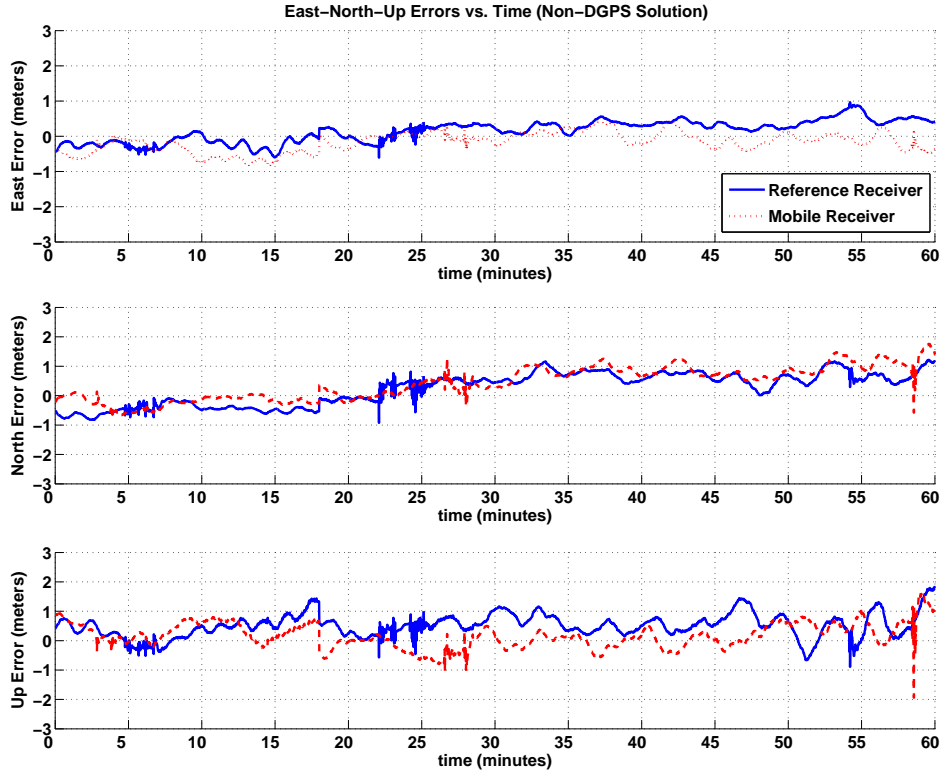


Figure 4.1: Static Testing, ENU Errors: Absolute Position (non-DGPS solution).

stand-alone (non-DGPS) navigation solution. Finally, one can see that the reference receiver errors are markedly similar to the mobile receiver errors.

A single-point DGPS relative position will now be formed that is nothing more than the difference between the two stand-alone absolute positions. Figure 4.2 compares the ENU errors of the two absolute positions (non-DGPS) to those for the single-point relative position (DGPS) for a data run of 75 minutes.

Although all three position solutions in Figure 4.2 are rather noisy, the benefit of DGPS is clearly evident. When looking at the first 30 minutes of the data run, both absolute positions have an Up error of approximate 1 meter. In contrast, the Up error of the relative position is quite small. In fact, for all three plots, the relative position errors are quite small when compared to the errors for the two absolute positions. This is a direct result of the error cancelation and/or minimization which

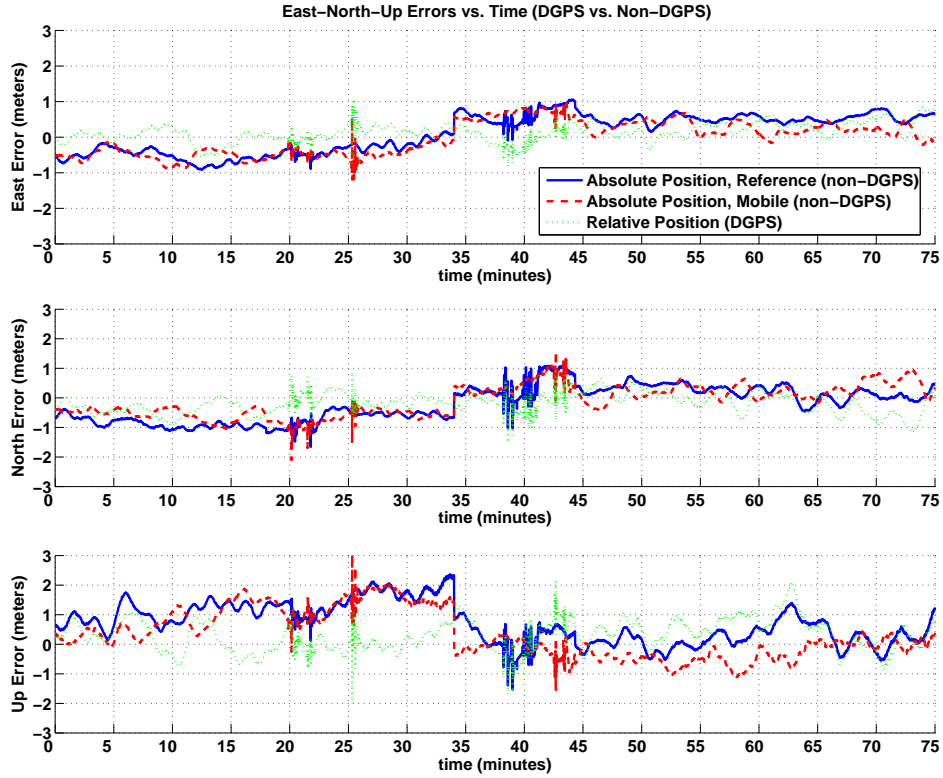


Figure 4.2: Static Testing, ENU Errors: Comparison Between Absolute Position (non-GPS) and Relative Position (DGPS).

can be expected when using Differential GPS. Of note, because both receivers were static, with a very short baseline, they shared the same SV geometry. As such, it is expected that the errors would be nearly identical.

Now that the benefit of utilizing DGPS, even if only by taking the difference between two absolute positions, has been demonstrated, the next section will focus on the static testing and analysis which was performed for the Kalman filter, ambiguity resolution, and high-rate output tasks.

#### 4.4 Static Test Results and Analysis

The results and analysis for static testing will now be presented. Recall that for static tests, the truth values for relative position were easily available.

*4.4.1 Kalman Filter Performance.* Because the Kalman filter provides estimates of the floating point double-difference carrier-phase ambiguities, and an associated covariance matrix, to the ambiguity resolution routine, it is important that the filter be properly tuned. If the Kalman filter error estimates, represented by the covariance matrix, are large when the filter estimates, represented by the state vector, are actually very good, then the performance of the filter will be degraded. Even worse is the case where the Kalman filter error estimates are very small, when in fact the filter estimates contain relatively large errors. Figure 4.3 shows the actual error in relative position (ECEF coordinates) for a data run of just over 5 minutes. Additionally, the state covariance matrix was used to provide the filter computed  $\pm 3\sigma$  standard deviation lines. The actual error should lie within the standard deviation lines 99 percent of the time.

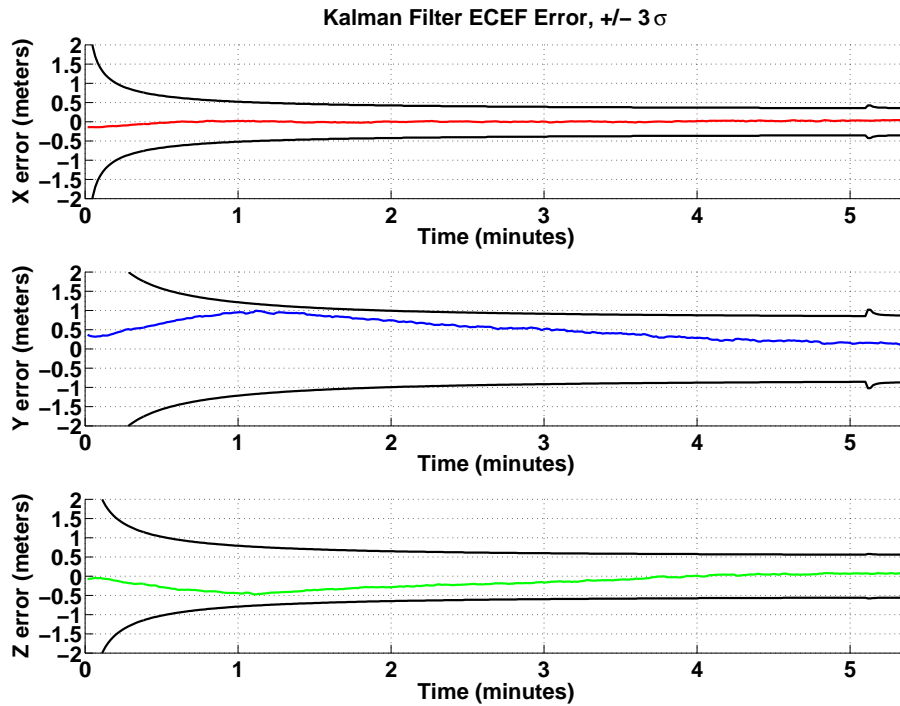


Figure 4.3: Static Testing, Comparison of Relative Position Error to Filter Computed Covariance Matrix (1 data run).

Clearly, the actual error lies within the error range expected by the filter. However, it must be noted that it is relatively easy to tune a Kalman filter for a specific set of data and achieve good results. Much more challenging is tuning the Kalman filter so that it provides good results for a variety of conditions. As such, and in an effort to conduct a more rigorous test of the performance of the Kalman filter, a test was conducted in which 50 different sets of data were collected. Each data set consisted of a 5 minute test run. Additionally, the data sets were collected at different times of the day and on different days of the week. By conducting the test in this manner, Kalman filter performance was tested using a different number of SVs and a variety of SV geometry conditions. Results for this test are presented in Figure 4.4. Note that the filter computed standard deviation lines vary from run to run. This is because the measurement matrix ( $\mathbf{H}$ ) changes from run to run (different line-of-sight vectors to each SV) and is used to propagate the state covariance matrix ( $\mathbf{P}$ ). As a result, the filter computed standard deviations are also different from run to run.

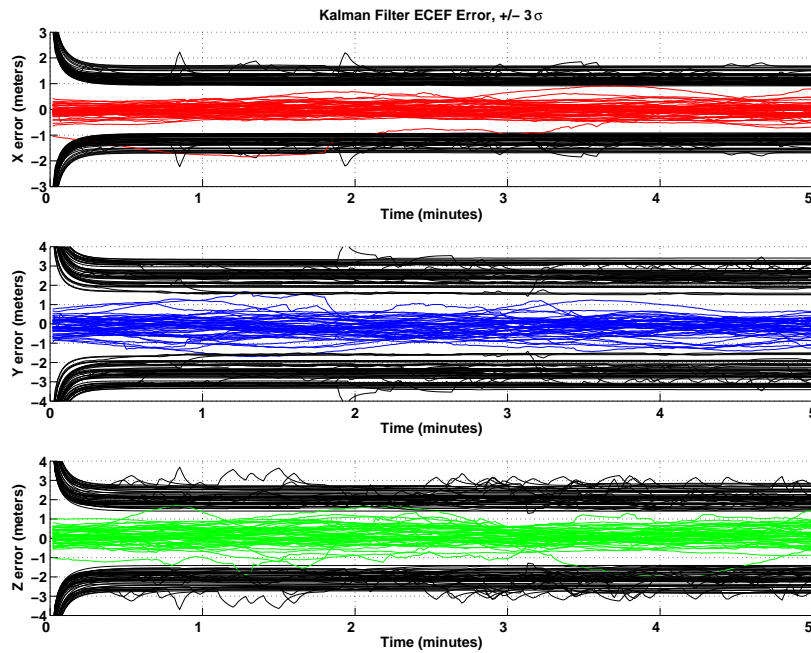


Figure 4.4: Static Testing, Comparison of Relative Position Error to Expected Error Range (50 data runs).

However, for the 50 data runs discussed above, the actual relative position error is consistently within the filter computed expected region, represented by the state covariance matrix, which is bounded by  $\pm 3$  standard deviations (99 percent).

The preceding discussion was for the state vector error analysis of relative position errors. However, a similar analysis can be performed for the floating point double-difference carrier-phase ambiguity errors. Before presenting the results for this test, a subtle difference between the two must be noted. In the case of relative position, the truth value for static testing will not change from one run to the next. However, in the case of the floating point estimate of carrier-phase ambiguities, the truth value will be different for each ambiguity pair during the same run. Additionally, the ambiguities from one run will generally be different from those found during another run. This is because the actual integer value of each of the double-difference carrier-phase ambiguities will generally be unique and can also vary from one run to the next. Figure 4.5 depicts the difference between the floating point double-difference carrier-phase ambiguity and the actual integer value.

Each of the four subplots presented in Figure 4.5 represent a separate run with a duration of 5 minutes. For a given subplot, each line represents the error for one of the double-difference carrier-phase floating point ambiguity estimates. As an example, during the run depicted in the upper left subplot, there were 6 SVs which resulted in an ambiguity set of 5 values, and 5 lines on the subplot. As before, lines of the filter computed  $\pm 3$  standard deviations are provided. One can see, as was the case for the relative position analysis, the actual error generally falls within the range expected by the Kalman filter. Additionally, as was previously discussed, because the measurement matrix changes from run to run, and is used to propagate the state covariance matrix, the lines representing  $\pm 3$  standard deviations will vary from run to run. This concludes the analysis of Kalman filter tuning. Attention will now focus on the role of the filter in dampening the noisy nature of the raw GPS measurements.

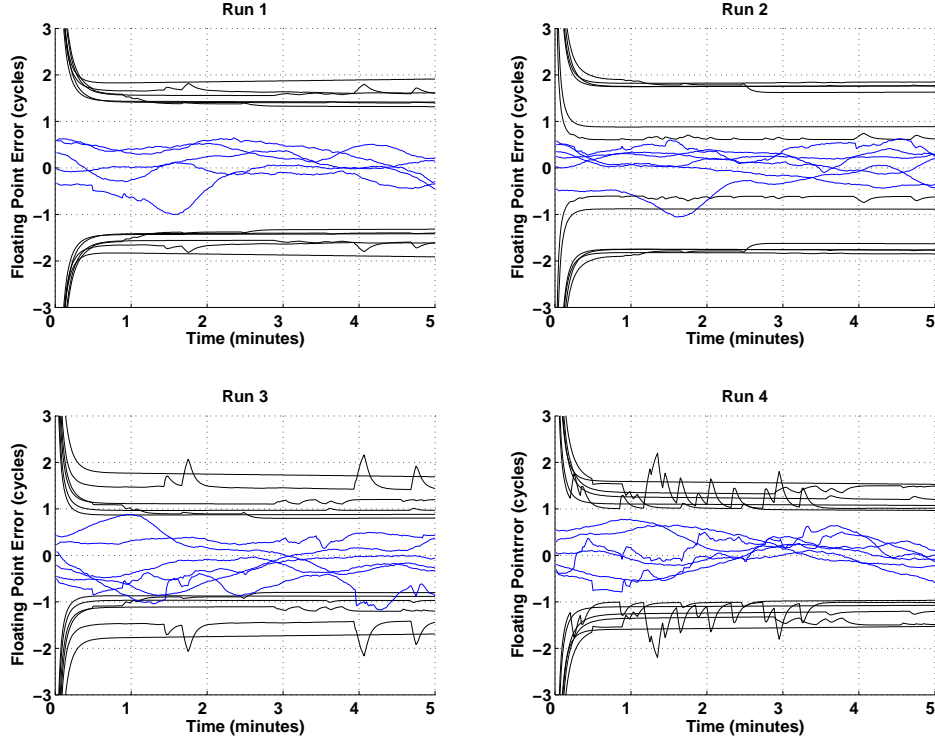


Figure 4.5: Static Testing, Comparison of Floating Point double-difference Carrier-Phase Ambiguity Errors to Expected Error Range (4 Separate Data Runs)

One of the key features of a Kalman filter is that it significantly dampens the noisy nature of the measurements while providing estimates of the state of a system. Figure 4.6 compares the East-North errors of the single point relative position, obtained by taking the difference between the NovAtel computed absolute positions, to the East-North errors of the Kalman filter estimated relative position.

The benefit of the Kalman filter is readily apparent. Whereas the relative position solution derived from the difference in the stand alone absolute positions (left side) is extremely noisy, the Kalman filter estimated relative position (right side) is much more stable. It must be noted the Kalman filter only utilized measurements that were common to both receivers. In contrast, the NovAtel computed absolute positions for each receiver utilized all of the SVs which were available to the respective receiver. The smooth nature of the filter relative position is due to both the



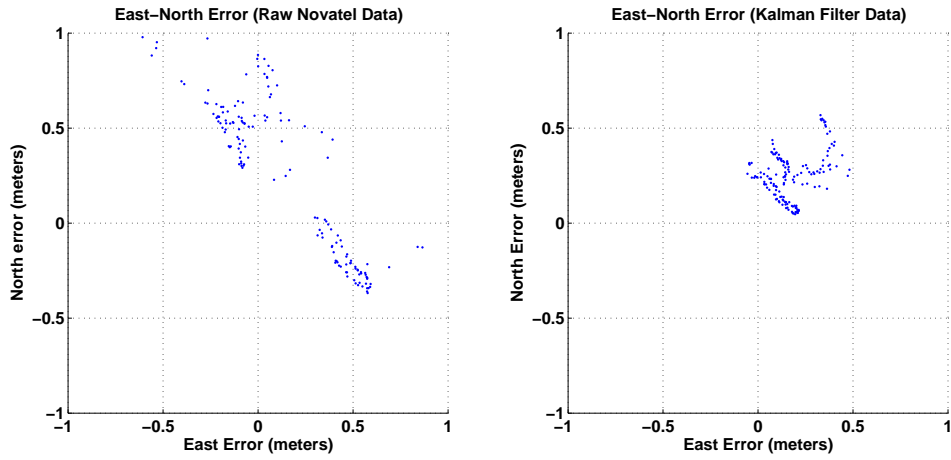


Figure 4.6: Static Testing, East-North Errors. Comparison of Filter Errors to Single Point Errors.

ability of the filter to dampen the raw GPS measurements and the use of common measurements.

Figure 4.7 presents similar data, but from a different test run. Additionally, the format has been changed to show the error for all three axes of the ENU coordinate system plotted versus time.

The value of utilizing the Kalman filter to dampen the noisy nature of the raw GPS measurements can clearly be seen. In the case of the single point NovAtel calculated relative position, the errors in all three axes exhibit large changes in magnitude over very short time spans. In stark contrast, the Kalman filter estimated relative position provides a much more stable solution.

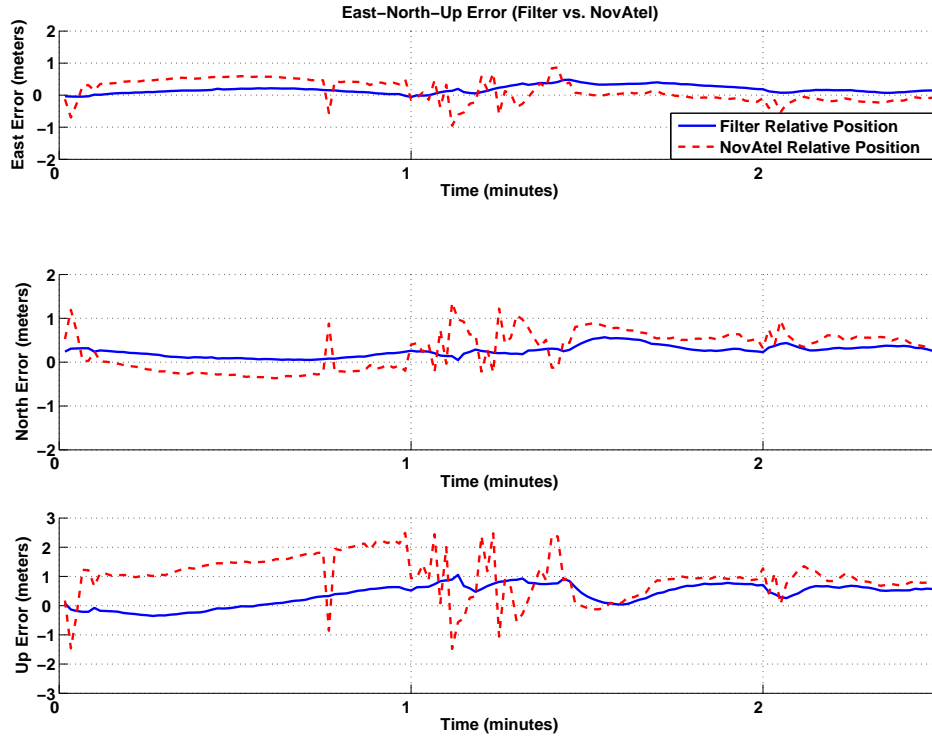


Figure 4.7: Static Testing, ENU Errors. Comparison of Filter Errors to Single Point Errors.

*4.4.2 Ambiguity Resolution Performance.* As mentioned in Chapter 3, the LAMBDA routine generates two possible double-difference carrier-phase integer ambiguity sets. Each ambiguity set has an associated squared normal value, described by Joosten [5], which is an approximation of how good the LAMBDA routine believes the set is. The reason for generating two sets is that although the correct ambiguity set will usually have the lowest value, sometimes it does not. An incorrect ambiguity set, for a brief time, may have a squared normal value that dips slightly below that of the correct set. Figure 4.8 provides a comparison of squared normal values for the “best” and “second best” ambiguity sets on four separate data runs. As will be discussed later in the section, after the ambiguities were solved for (generally 2 seconds), the filter was re-initialized and a new floating point estimate and covariance matrix were passed to the ambiguity task. In this mode, a new squared normal value was

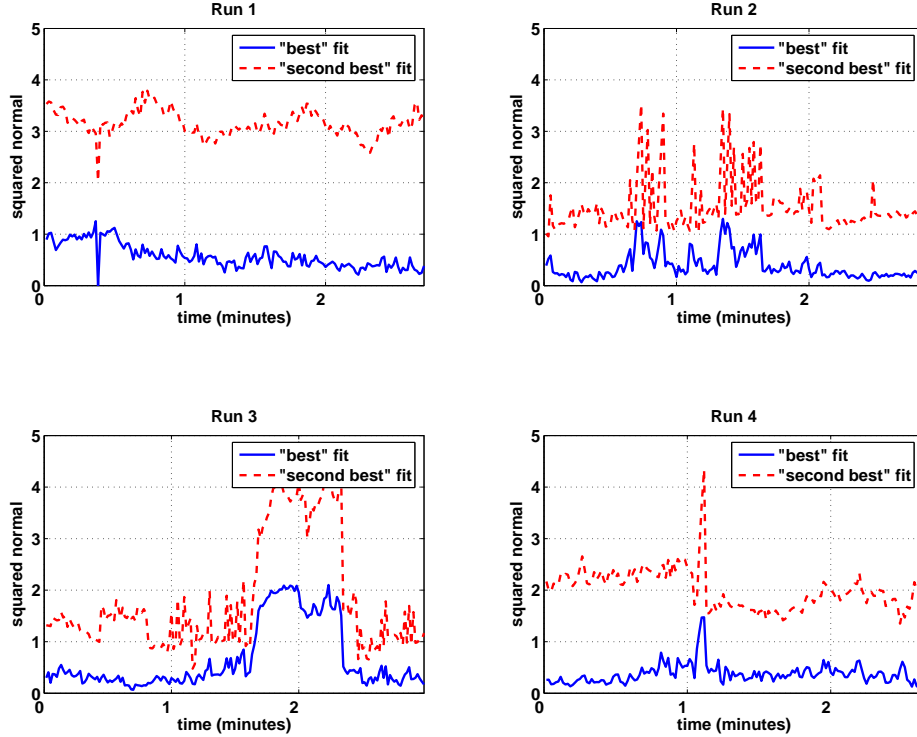


Figure 4.8: Static Testing, Comparison of Squared Normal Values. Four Data Runs (Filter Reset)

computed approximately every 2 seconds. The system does not usually operate in this mode; however, it was useful in obtaining a large number of observations regarding the algorithm's ability to determine the ambiguity set correctly and quickly.

By far, the most common trend observed during testing is that shown in the upper left quadrant of Figure 4.8. In this case, the “best” solution has a squared normal value that is significantly lower than that of the “second best” value. In this case, the correct ambiguity set can be chosen very quickly and with high confidence. As mentioned, this case was by far the most common one seen during testing. However, several other types of behavior were observed and these will now be discussed. The data run shown in the upper right quadrant shows a time segment where both the “best” and “second best” squared normal values exhibit significant fluctuations for an extended time. Another type of behavior that was observed during testing is

shown in the lower left quadrant. In this case, besides having fluctuations, both the “best” and “second best” squared normal values exhibit a single very large jump and remain at an elevated level before returning to lower values. In this case, the critical value of 2.5 was exceeded for almost the entire run. The final type of behavior that was seen during testing is shown in the lower right quadrant. This case is similar to the one just discussed. However, the difference is that the values did not remain at elevated levels after a significant jump. Instead, they returned very quickly to the lower levels that existed immediately before the jump.

In the preceding discussion, the Kalman filter was reset immediately after the critical value was exceeded. However, it is also beneficial to look at the characteristics of the squared normal values when the Kalman filter is not reset even though the ambiguities are calculated at every epoch. Figure 4.9 shows results from four separate data runs where the Kalman filter operated continuously but the ambiguities were calculated every epoch.

Because the Kalman filter was not reset, and instead allowed to operate continuously, the floating point estimates and covariance matrix passed to the ambiguity resolution routine become more and more accurate. As a result, the squared normal value of the “best” set generally starts low and stays low. In contrast, the squared normal of the “second best” set diverges quickly which makes ambiguity resolution relatively easy. However, as can be seen in the upper right quadrant, even with the filter operating continuously, there were times where the difference between the two values converge.

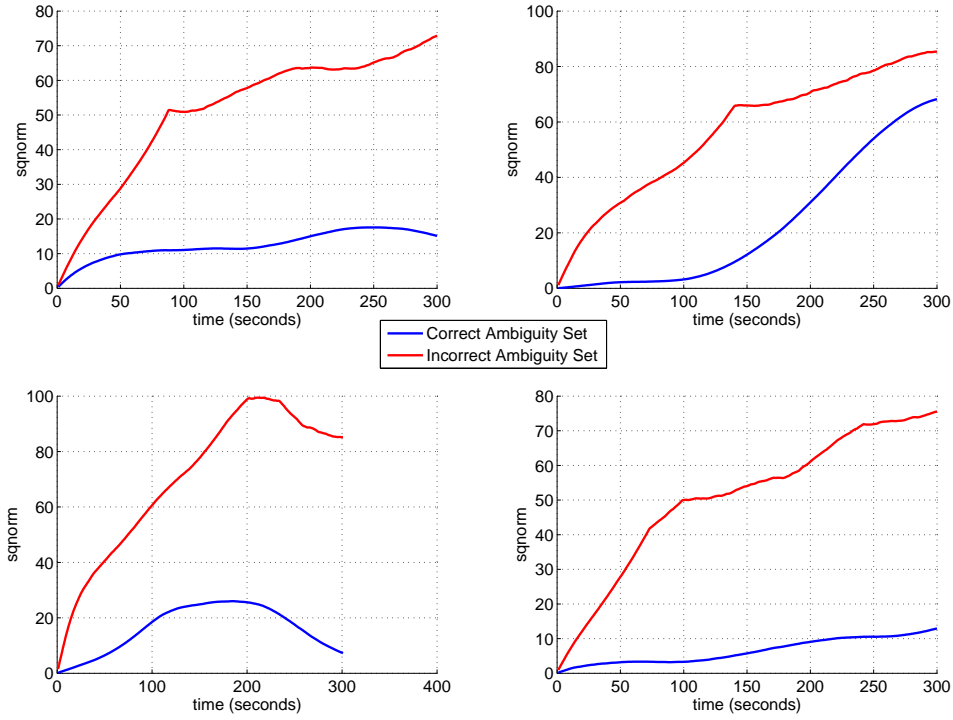


Figure 4.9: Static Testing, Comparison of Squared Normal Values. Four Data Runs (No Filter Reset)

As discussed in Chapter 3, to ensure the correct ambiguity set is chosen, the ambiguity resolution task utilizes a ratio test with a critical value of 2.5 and waits until this threshold is exceeded before passing the ambiguity set to the high-rate output task. In order to determine the time required to resolve the ambiguities, and the success rate of choosing the correct ambiguity set, a test was conducted which is discussed below.

The raw measurements for all static testing were re-run in post-processed mode with several variations to the program. First, after exceeding the critical value, the number of epochs from filter initialization to solution were saved. Additionally, the ambiguity set associated with the “best” squared normal value was saved. Finally, the correct ambiguity set, analytically determined as discussed in Chapter 3, was saved. A simple comparison was made between the computed set and the correct set. If the

two sets were identical, the instrumentation package chose the correct set. If the two sets were different, then the wrong ambiguity set was chosen. The results of this test are provided in Table 4.2 below.

Table 4.2: Ambiguity Resolution Performance (Widelane Measurements)

Time to Solve Ambiguities	Occurrences	Percentage
2 seconds	6,750	98.1 percent
3 seconds	78	1.1 percent
4 seconds	23	0.4 percent
5 seconds	12	0.2 percent
6+ seconds	13	0.2 percent

The longest time to reach the critical value was just under two minutes (117 seconds). However, most of the longer convergence times were much shorter in duration.

In terms of success rates, for the 6,876 calls to the ambiguity resolution routine, there were only 7 instances in which the returned ambiguity set was incorrect. This results in a 99.9 percent success rate. Achieving 100 percent success rate is possible if a critical value above 2.5 is used. However, selecting a higher critical value can result in substantially longer times to converge to the correct ambiguity set. Raising the critical value to 5.0 resulted in numerous instances of a 3-4 minute delay before the ratio test passed. Additionally, the few ambiguity sets that were wrong were not drastically wrong. Most had only one integer ambiguity that had been rounded the wrong way. As a result, the relative position error achieved when using the wrong ambiguity set was on the order of 1-2 meters. Although this is excessive for precise positioning, it was deemed acceptable given the extremely low probability of occurrence.

As will be seen in the next section, the overall accuracy obtained when utilizing widelane measurements is relatively poor. The accuracy can be significantly improved if the widelane measurements are replaced with either L1 only measurements or narrowlane measurements. However, ambiguity resolution becomes more difficult when widelane measurements are not used. To demonstrate this, the test discussed above was repeated using the same raw measurements, but instead of using

widelane measurements, the test utilized L1 only measurements. The results of this test are provided in Table 4.3 below.

Table 4.3: Ambiguity Resolution Performance (L1 Measurements)

Time to Solve Ambiguities	Occurrences	Percentage
2 seconds	2,978	93.5 percent
3 seconds	86	2.7 percent
4 seconds	32	1.0 percent
5 seconds	23	0.7 percent
6+ seconds	65	2.0 percent

In terms of success rates, for the 3,090 calls to the ambiguity resolution routine, there were 94 instances in which the returned ambiguity set was incorrect. This results in a 97.0 percent success rate. Although these results are not bad, they are noticeably lower than the 99.9 percent success rate obtained with the use of widelane measurements. Additionally, in terms of convergence times, of the 65 occurrences that were six seconds or longer, most were several minutes in duration. When narrowlane measurements were used, results for the above test were significantly worse. Thus, in terms of ambiguity resolution, the use of widelane measurements offers the best performance. As a result, in order to obtain the desired accuracy, the high-rate output task will have to transition from widelane to L1 only and finally to narrowlane measurements. This, along with other considerations, will be discussed in the next section.

*4.4.3 High-Rate Output Performance.* A key measure of performance for the instrumentation package is the precision of the high-rate output. Because the truth value for static testing was known, a simple comparison between the actual relative position and the reported relative position is possible. Figure 4.10 provides two views of the high-rate relative position errors over a 5 minute test run. The upper figure is a top-down view which shows the North and East errors. The lower figure is a forward looking view which shows the Up and East errors.

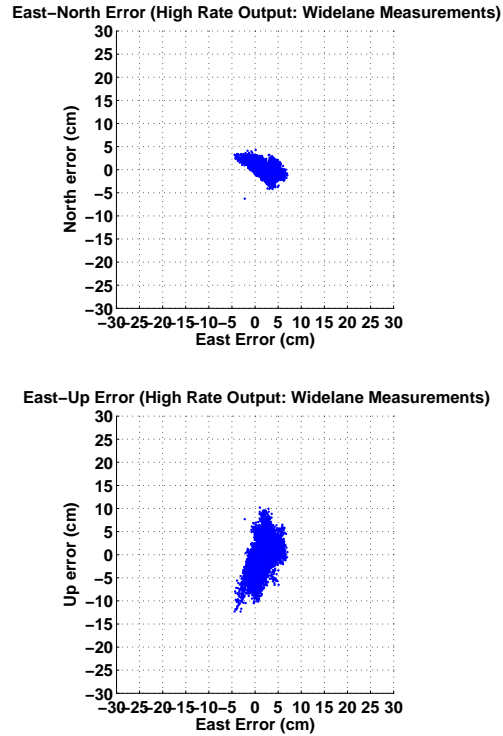


Figure 4.10: Static Testing, high-rate Output: Comparison of Horizontal Errors and Vertical Errors (Widelane Measurements)

An inspection of Figure 4.10 yields an initial estimated horizontal accuracy of approximately  $\pm 5$  cm and an estimated vertical accuracy of approximately  $\pm 10$  cm. The tendency for the vertical error to be roughly twice the horizontal error was found in virtually all data runs and is consistent with what is commonly observed in GPS and DGPS positioning.

The relative concentration of points shown in Figure 4.10 was found in almost all of the data runs. However, many of the runs exhibited biases in one or more directions. Figure 4.11 shows the horizontal and vertical errors for several different data runs superimposed on top of each other. Each data run is represented by a different color.

One can see that the relative position solution appears to wander over time and the overall precision achieved is undesirable. After ensuring that the apparent



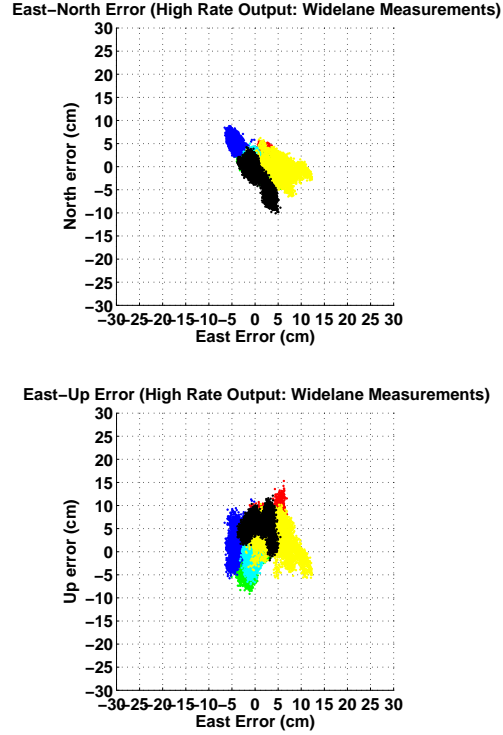


Figure 4.11: Static Testing, high-rate Output: Biases Found in ENU Errors (Wide-lane Measurements)

drift was not due to tropospheric error or differences in clock error between the two receivers, attention was turned to multipath as a possible cause. Two tests were conducted to ensure that the errors found during initial testing were in fact due to multipath and not other factors. The first test consisted of collecting multiple data sets at the same time interval on two consecutive days. The second test consisted of computing the relative position with L1 only measurements instead of wide-lane measurements. Both of these tests, and their associated results, are discussed below.

Because of the periodic nature of the SV orbits, one can replicate the exact SV geometry by waiting exactly 24 hours and 4 minutes. Additionally, for static testing, the error caused by multipath will be nearly the same for two observation periods having the same SV geometry, because multipath is dependent on the relative geometry between the satellites and the receiver. Therefore, if the error causing the

apparent wander discussed above is truly due to multipath, and not due to other factors, then one should obtain the same error biases for multiple runs taken at intervals of 24 hours and 4 minutes.

Figure 4.12 shows the relative position error (ECEF coordinates) versus time for two runs taken at 24 hour intervals. One can see that the magnitude of the errors for the runs shown are similar. However, it is difficult to determine the extent to which the run taken on Thursday is offset relative to the run taken on Wednesday. Therefore, a second graph was generated in which the measurements for Thursday were shifted to the right by exactly 4 minutes. If the error causing the wander discussed above is really due to multipath, then the errors should have a high degree of correlation. Figure 4.13 provides the the graph with the above mentioned offset applied. It can easily be seen that the correlation between the two runs, with the 4 minute offset added in, is almost perfect. As a result, most of the apparent drift observed during testing is almost certainly due to multipath.

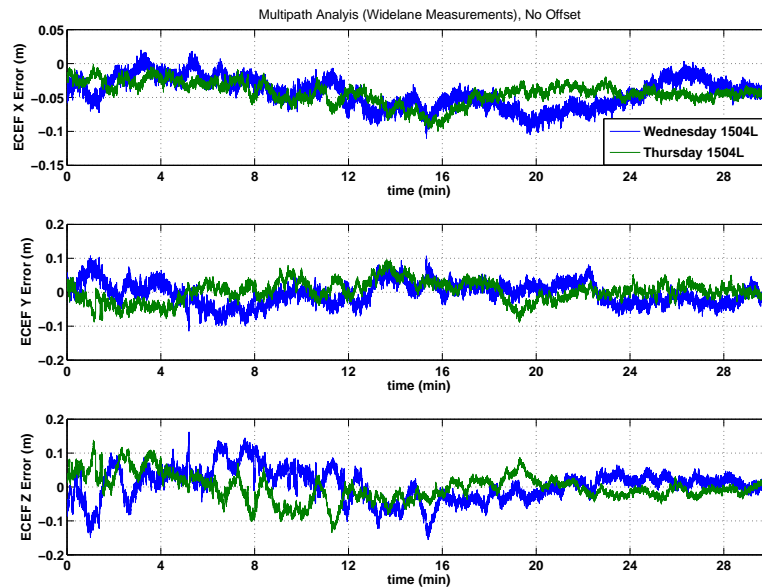


Figure 4.12: Static Testing, High-Rate Output: Periodicity of Multipath Errors (No Offset)

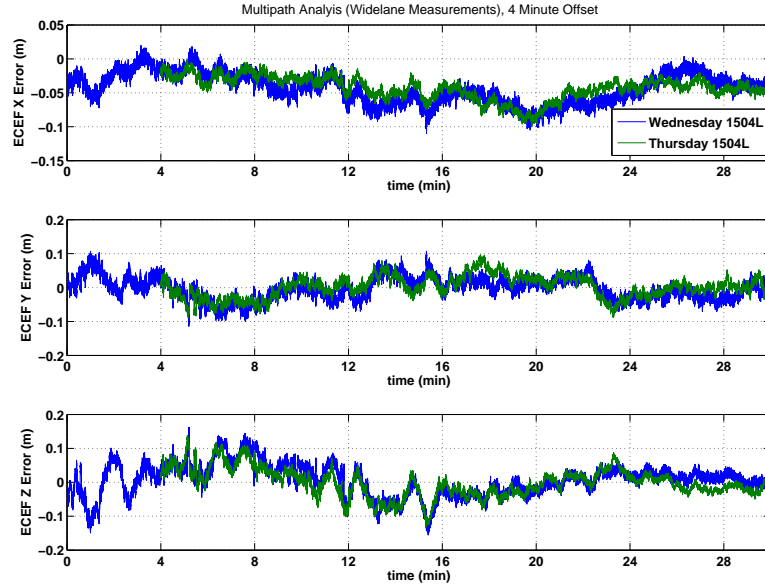


Figure 4.13: Static Testing, High-Rate Output: Periodicity of Multipath Errors (4 Minute Offset)

Another independent means of determining whether the apparent drift was caused by multipath is to compare the relative position solution obtained using widelane measurements to that obtained when using L1 only measurements. This is because the errors caused by multipath will be reduced by a factor of approximately four when switching from widelane measurements to L1 only measurements [15]. Hence, if the accuracy is generally four times better when using L1 only measurements than for the case when widelane measurements are used, then the apparent drift discussed above is likely due to multipath, and not other factors.

Figure 4.14 provides a comparison of the accuracy obtained by using widelane measurements to that obtained when utilizing L1 measurements for the same data run. The top two plots depict the relative position error when widelane measurements were used. The bottom two plots depict the relative position error when L1 only measurements were used. These results also provide a strong indication that the apparent drift observed during initial widelane measurement testing is in fact due to multipath errors.

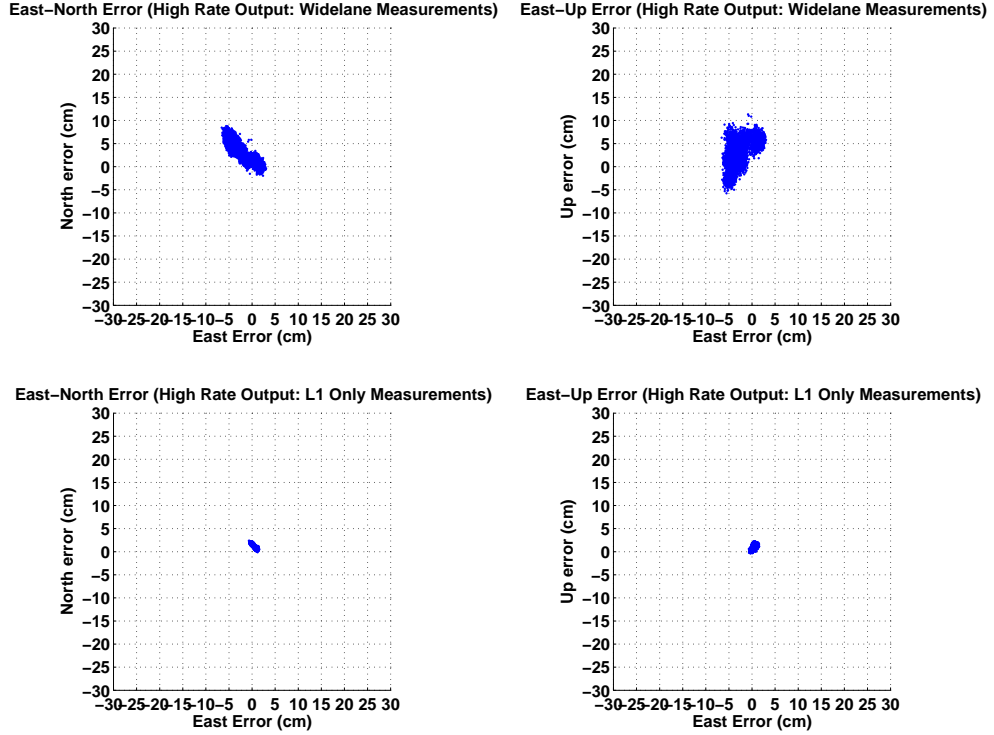


Figure 4.14: Static Testing, High-Rate Output: Comparison of Accuracy (Wideline Measurements vs. L1 only Measurements)

Because multipath errors are significantly higher when using widelane measurements, combined with the better accuracy obtained when using L1 only measurements, one might wonder why widelane is used at all. The answer lies in ambiguity resolution. The benefit of higher accuracy obtained by using the lower wavelength of L1 measurements is offset by the added difficulty in determining the correct ambiguity set. With the critical value left at 2.5 as was discussed in the previous section, there were instances where it took in excess of four minutes for the ambiguity resolution task to determine the ambiguities when L1 only measurements were used. Additionally, even after the ratio test passed, the success rate of choosing the correct ambiguity set was noticeably lower when L1 only measurements were used than was found when widelane measurements were used. For this reason, the instrumentation package presented in this thesis utilizes widelane measurements until the ambiguities are resolved.

Once the widelane ambiguities are known, an accurate relative position can be formulated. This newly formed position can then be used to analytically compute the correct ambiguities for L1 only measurements. The high-rate relative position, with better precision and accuracy, can then be recalculated using L1 only measurements and their associated integer ambiguities. This same procedure can be repeated to yield the correct ambiguities for narrowlane measurements. Of note, this process only has to be completed one time. After the L1 only, or narrowlane, ambiguities have been determined, the high-rate task continues in either L1 only mode or narrowlane mode as appropriate. The instrumentation package developed in this thesis has the capability to step from widelane measurements to L1 only measurements and finally to narrowlane measurements. However, time did not permit adequate testing of this process. One concern is whether the relative position solution obtained in L1 only mode is precise enough to correctly determine the integer ambiguities for narrowlane mode. This will be further discussed in Chapter 5.

A statistical analysis was conducted to compare the precision of widelane, L1 only, and narrowlane measurements. Following a data collection period of 40 minutes, the same raw measurements were run three times in post-processed mode. During the first run, the instrumentation package utilized widelane measurements exclusively. During the second run, L1 only measurements were used. During the third run, narrowlane measurements were used. Of note, all relative positions were converted to ENU coordinates. Table 4.4 provides the results of the statistical analysis.

One of the first items in Table 4.4 that draws attention is the relatively large error bias of over 3 cm in the East/West direction for each of the three modes. It is believed that this bias is due to an error in the “truth” values. Recall that the truth values were obtained by taking the difference between the two absolute positions returned from OPUS. If these positions are slightly inaccurate, then the computed truth value will be inaccurate as well. Fortunately, in addition to the calculated absolute value, OPUS returns standard deviation values as well. The standard deviations for the OPUS computed “truth values” are shown in Table 4.5.

Table 4.4: Statistical Analysis of High-Rate Output (ENU Coordinates)

	East	North	Up
Truth Value	-2.365	5.645	-0.792
Average Relative Position (Widelane)	-2.397 m	5.631 m	-0.793 m
Average Relative Position (L1 only)	-2.401 m	5.647 m	-0.796 m
Average Relative Position (Narrowlane)	-2.401 m	5.649 m	-0.797 m
Mean Error (Widelane)	-3.22 cm	-1.49 cm	-0.10 cm
Mean Error (L1 only)	-3.60 cm	0.27 cm	-0.42 cm
Mean Error (Narrowlane)	-3.65 cm	0.49 cm	-0.46cm
Standard Deviation (Widelane)	1.03 cm	2.12 cm	4.17 cm
Standard Deviation (L1 only)	0.19 cm	0.49 cm	0.77 cm
Standard Deviation (Narrowlane)	0.15 cm	0.39 cm	0.69 cm

Table 4.5: Standard Deviation for OPUS Truth Values (ENU Coordinate Frame)

	East	North	Up
Reference Receiver	2.05 cm	1.35 cm	0.69 cm
Mobile Receiver	2.88 cm	1.88 cm	0.53 cm

Thus, the standard deviation of the East/West component returned from OPUS is large enough to indicate that the relative position truth value may be a centimeter or two off from the actual value. Table 4.6 provides the computed DRMS values for the same data run just discussed.

Table 4.6: Static Testing: Typical DRMS Values (ENU Coordinate Frame)

	DRMS
Widelane	4.26 cm
L1 only	3.65 cm
Narrowlane	3.71 cm

It can be seen that making the transition from widelane measurements to L1 only measurements yields significant improvements. The transition from L1 only to narrowlane, at least in the static environment, does not add much in the way of performance. However, the computational overhead of switching between modes is extremely small. As a result, the instrumentation package presented in this thesis converges to the use of narrowlane measurements.

In addition to precision, a key performance criterion is to minimize system latency. For purposes of discussion, system latency is defined as the length of time that passes from the time the instrumentation package receives a RANGECMPB data log from each of the two serial ports until the time that the use of those measurements result in a computed relative position vector. To clarify, the amount of time required to read the incoming data log from each serial port and identify them both as being a RANGECMPB log is not included in the latency computations. However, the system latency does include the time required by the COM 1 and COM 2 threads to decode the binary log, the time required by the parser to synchronize the measurements, and the time required by the high-rate output task to collect data from the Kalman filter task, the ambiguity resolution task (if required), and compute the relative position. Figure 4.15 shows the system latency for a sample data run of just under seven minutes.

One can see that the overall system latency is relatively steady at 10 ms with a brief spike of 15 ms. In order to have the capability to process data at a rate of 100 Hertz, the system latency must never exceed 10 ms. It would appear initially that system latency would prohibit operation at 100 Hertz, even if the data link could handle the required throughput of approximately 400 Kbps. However, by examining a much shorter sample period, the system latency can be reassessed. Figure 4.16 shows the system latency for a small portion of the same run which was depicted in Figure 4.15.

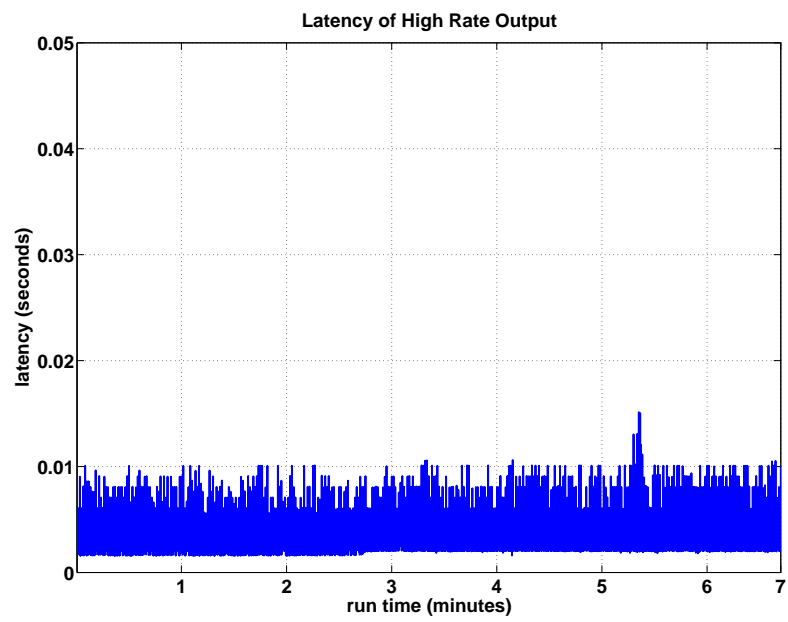


Figure 4.15: Static Testing - System Latency

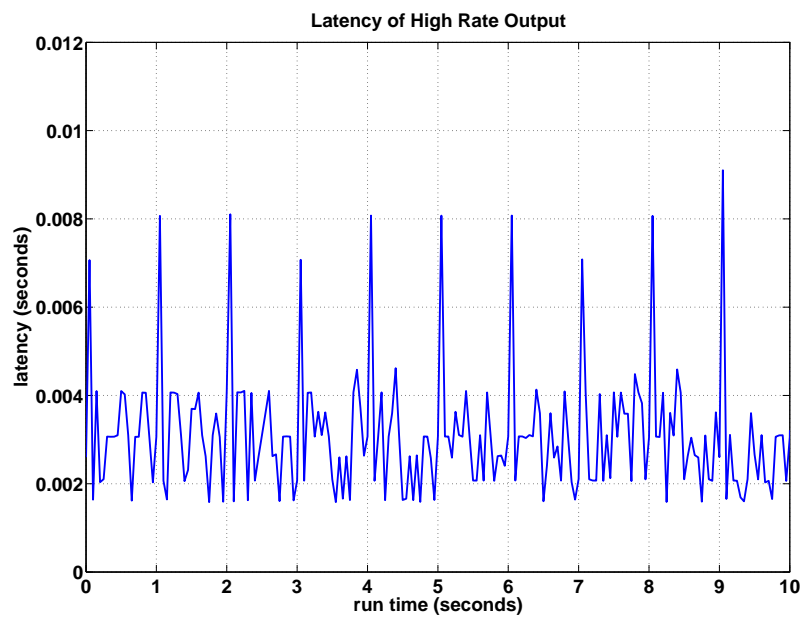


Figure 4.16: Static Testing - System Latency, Expanded View



One can see that the steady-state latency of the high-rate output is between 2-4 ms. This is considerably less than was evident in Figure 4.15. The spikes in total latency, which reveal a latency of 8-10 ms is clearly due to the added processing of the Kalman filter which operates at a 1 Hertz rate. Thus, there are two ways to improve performance and be able to provide 100 Hertz output. The first option is to use a CPU which is faster than the 1.4 GHz processor with a 400 Hz clock speed. However, another possibility is to adjust the priority of the threads to give precedence to the high-rate output. Because the Kalman filter has nearly a full second to process its data, it can be given a much lower priority than the high-rate output which has much less time to complete its required computations. It should be noted that selecting one of the two previously mentioned options is only required if 100 Hertz output is required, or if a less capable processor is used. The second case is rather likely for a flight test scenario because of size, weight, and power consumption restrictions. Thread prioritization is discussed in Chapter 5. For 20 Hertz operation, the instrumentation package presented in this thesis does not need to be adjusted.

Just as important as low latency, the high-rate output must provide relative position solutions at a steady rate. For use in formation flight control, where inputs are seen as step inputs, the output interval must be consistent. Figure 4.17 shows the interval between computed relative position solutions for a data run of just under seven minutes.

As can be seen, the high-rate output generates a relative position solution that is generally within 10 ms of the ideal 20 Hertz rate. Occasionally, the interval is slightly more or slightly less. However, it must be noted that the times were gathered directly from the system clock time, rather than a high-precision timer. This was done because of the inherent difficulty of incorporating a high-precision timer into an application which is simultaneously running multiple threads asynchronously. This concludes the static testing and analysis. We will now consider the dynamic scenarios which were examined.

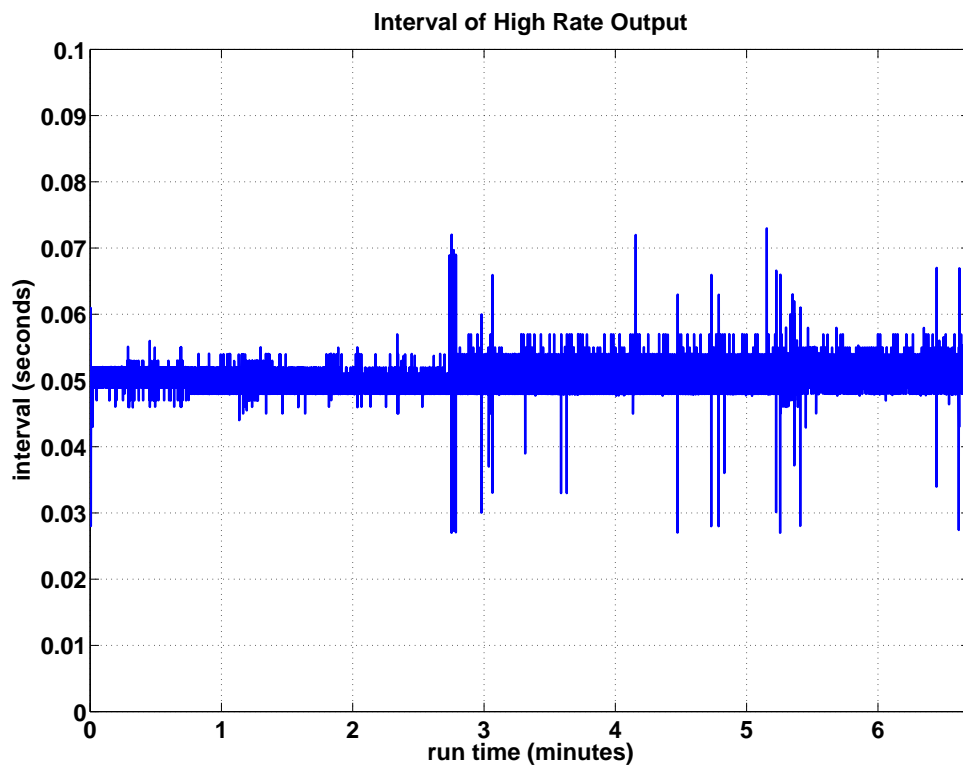


Figure 4.17: Static Testing - Interval of Relative Position Solution

#### ***4.5 Dynamic Test Results and Analysis***

As previously discussed, truth values for dynamic testing are more difficult than for the static environment. Dynamic testing was conducted utilizing a modified golf cart which contained two GPS receivers, two GPS antennas, two CPU's, and two wireless serial data link transceivers. Additionally, a base reference station consisted of two GPS receivers, a common antenna, and two wireless serial data link transceivers. One pair was dedicated to the instrumentation package presented in this thesis. The second, independent pair, was dedicated to a DGPS application from NovAtel.

The instrumentation package logged the real-time relative position to a data file. The truth values were obtained from the NovAtel GPS solution software mentioned above. Of note, the NovAtel system cannot provide real-time relative position. Instead, it provides real-time absolute position with centimeter-level accuracy. It accomplishes this through the use of a GPS reference station and a wireless serial data link which provides differential corrections. By taking the difference between the NovAtel computed absolute position and the absolute coordinates of the reference station, a post-processed relative solution was formulated. This relative position truth value was then compared to the real-time relative position obtained by the instrumentation package.

One complication that arose during testing was that the mobile location had to use two closely spaced antennas, one for each receiver, rather than a common antenna for both. This resulted in a distance of 24 cm between the two antennas. Depending on the heading of the golf cart, this offset was applied in a different cardinal direction.

Figure 4.18 provides a comparison of the post-processed horizontal relative position computed by the Novatel DGPS application to the real-time horizontal relative position computed by the system presented in this thesis. Although the offset due to the use of separate antennas is obvious, the results clearly show that the two systems computed nearly identical tracks over the ground.

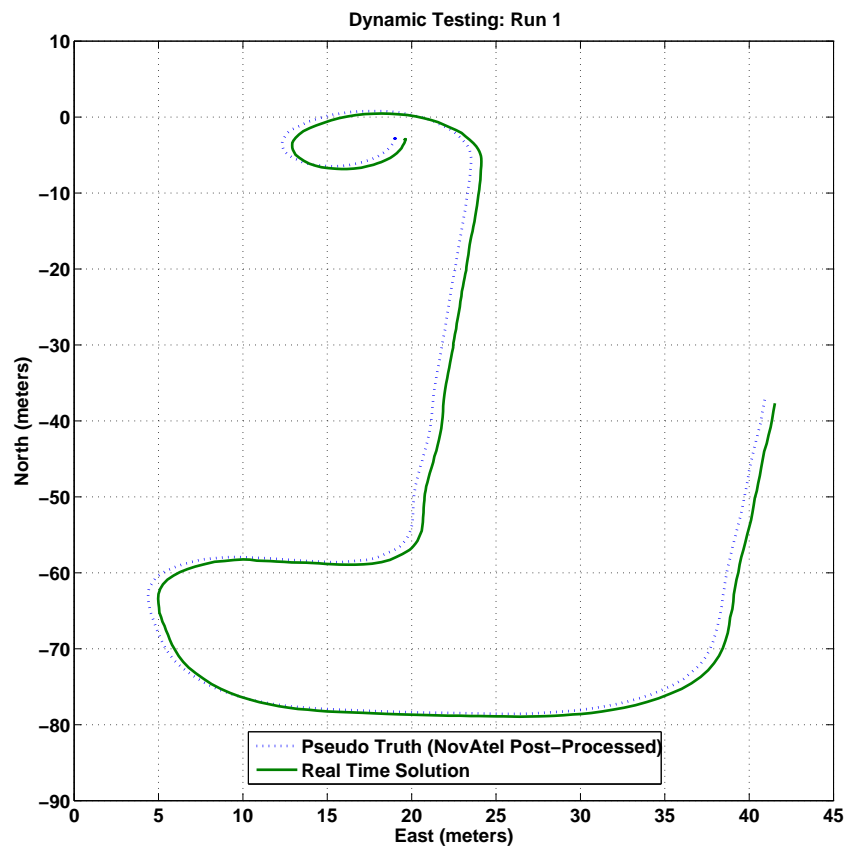


Figure 4.18: Dynamic Testing - Horizontal Relative Position

As mentioned in Chapter 3, the Kalman filter gives the user the option of obtaining relative velocities and relative acceleration vectors as well as relative position. For obvious reasons, during static testing, this capability was not discussed. However, for dynamic testing, it is worthy of discussion.

Figure 4.19 and Figure 4.20 provide the Kalman filter relative position and relative velocity plotted versus time for the same data run. For ease of discussion, let's only focus on the X axis (top third of each subplot). Additionally, let's focus attention on the series of S-turns beginning 48 seconds into the run and ending 75 seconds into the run. From the velocity plot, it can be seen that the golf cart began a turn to the left. The position plot confirms a displacement to the left. After a delay of approximately 7 seconds, the golf cart reversed directions to the right. Both the velocity plot and the position plot confirm a reversal took place. Finally, at approximately 65 seconds into the run, the golf cart reversed directions (to the left) and returned to its original heading. Both the velocity plot and position plot confirm these dynamics. Additionally, an inspection of both plots reveals that the magnitudes of the velocities are consistent with the magnitude of the displacements in position. Of note, this discussion has focused on the Kalman filter computed relative position and relative velocity terms. A similar analysis could be performed for relative acceleration but would yield the same results.

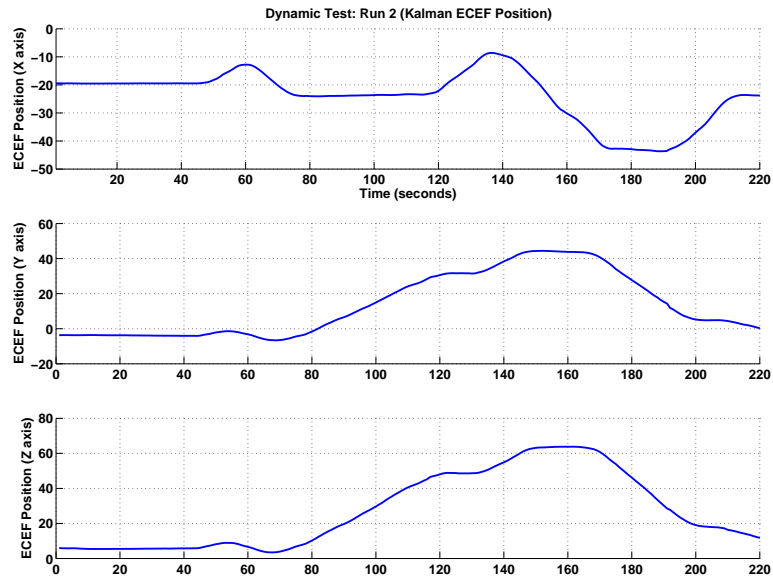


Figure 4.19: Dynamic Testing - Kalman Filter: Relative Position

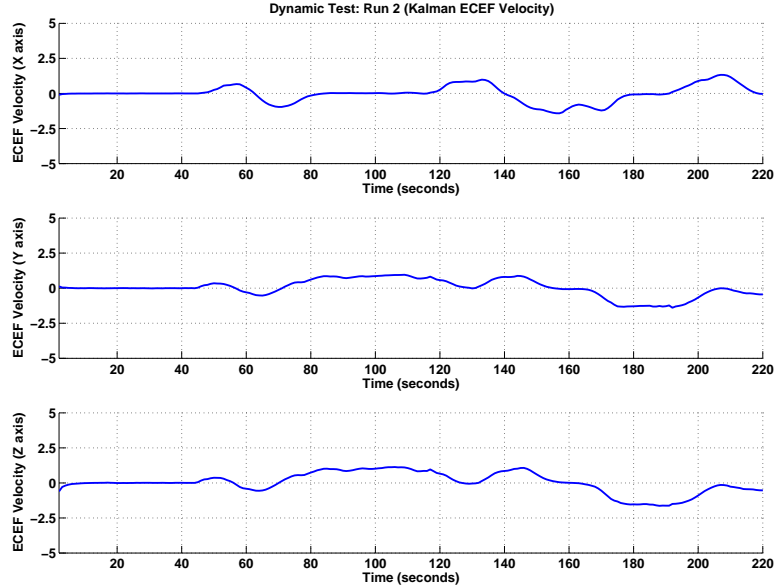


Figure 4.20: Dynamic Testing - Kalman Filter: Relative Velocity

## 4.6 *Asynchronous Input Analysis*

A final consideration is the determination of the impact of having the Kalman filter task and ambiguity resolution task provide data to the high-rate output task at rate of less than 20 Hertz. In the case of the ambiguity resolution task, this determination is quite easy. The integer ambiguity values determined utilizing the LAMBDA method and the ratio test are only passed to the high-rate task one time. After this occurs, the high-rate output task can use the known ambiguities to determine the relative position and can use the known position to calculate any remaining unknown ambiguities. However, the Kalman filter passes the unit line-of-sight vector information for each SV to the high-rate task at a reduced 20 Hertz rate. This results in up to a second of latency in the reduced measurement ( $\mathbf{H}$ ) matrix and is worthy of follow-on analysis.

Figure 4.21 shows the ECEF errors in relative position for a very short time span of 5 seconds. If the latency associated with a 1 Hertz update rate of the measurement matrix were a concern, then there would be apparent step inputs to the graph. However, since the graph is relatively smooth from one second to the next, the very small errors that are induced by using a slightly outdated  $\mathbf{H}$  matrix are negligible. Given the large distances involved, it is expected that the unit line-of-sight vectors will change very slowly. Hence, this is why the Kalman filter was designed to operate at a much slower rate to save computational overhead.

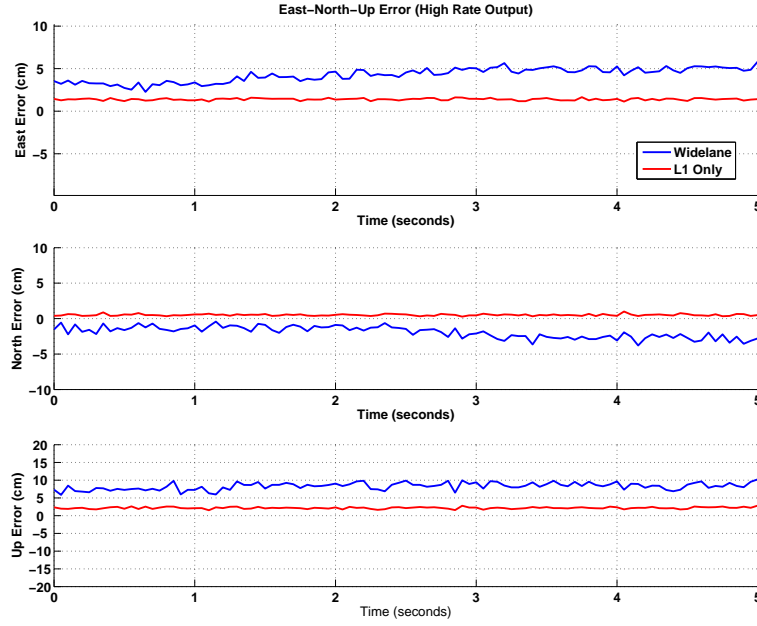


Figure 4.21: Effect of Asynchronous Data: ECEF Errors (Expanded View)

#### 4.7 Summary

This chapter has presented the results and analysis for both static and dynamic ground testing. In terms of Kalman filter performance, attention was placed on filter tuning and the ability of the filter to dampen the noisy nature of the raw GPS measurements. Additionally, the capability of the filter to provide relative velocity and relative acceleration in a dynamic environment was shown. For the ambiguity resolution task, attention was placed on the relationship between the squared normal value associated with the “best” ambiguity set and the “second best” ambiguity set. Additionally, the time to select the correct ambiguity set and the success rate of actually choosing the correct set were discussed. In terms of the high-rate task, both the precision of the relative position and the system latency were evaluated.



## V. Conclusions and Recommendations

This chapter provides a brief summary of the system design and highlights the results of the testing and analysis which was conducted. Additionally, recommendations are made that will hopefully be beneficial to those who elect to add to the research effort in this area.

### 5.1 Overview

The goal of the research presented in this thesis was to develop a DGPS instrumentation package capable of providing high-rate, extremely precise, relative position solutions between two small UAVs, with very low latency. Because the system was designed for small UAVs, a key criteria was to minimize the overall size, weight, and power consumption.

The raw GPS measurements from the lead UAV (reference receiver) are transmitted to the wing UAV via a wireless serial data link. The raw measurements from the wing UAV (mobile receiver) are sent to the CPU by means of a serial cable connected to a COM port. The CPU, onboard the wing aircraft utilizes multi-threading to simultaneously decode both binary streams into a useable format and pass the data to the parser discussed below.

Because DGPS applications require common measurements between the reference receiver and the mobile receiver, a front-end parser is used to strip off any measurements that are not common to both the lead and wing UAV's. The data that is common to both UAV's is passed at a 1 Hertz rate to a Kalman filter and at a 20 Hertz rate to a high-rate output task.

The core of the system consists of three tasks which operate asynchronously yet share information when required. These tasks consist of a Kalman filter, an ambiguity resolution routine, and a high-rate output task. A brief discussion of each is provided below.

The Kalman filter task, operating at a 1 Hertz rate, receives raw GPS measurements from the parser and provides several useful outputs. First, a low-rate,

approximate relative position solution is provided. Additionally, the user has the option of receiving relative velocity and relative acceleration estimates as well. Second, the Kalman filter provides a floating point estimate of the double-difference carrier-phase ambiguities, and an associated covariance matrix, to the ambiguity resolution routine when required. Third, the Kalman filter calculates the unit line-of-sight vector from the mobile UAV to each SV in view.

The ambiguity resolution routine is summoned by the Kalman filter whenever the floating point estimates of the double-difference carrier-phase ambiguities need to be resolved to their integer values. It receives the floating point estimates, and an associated covariance matrix, from the Kalman filter and utilizes the LAMBDA method to resolve the ambiguities. Because the “best” set may not necessarily be the correct set, the LAMBDA routine returns two possible ambiguity sets with an associated squared normal value which represents how good the LAMBDA routine believes the fit is for each ambiguity set. The ambiguity resolution routine then utilizes a ratio test with a critical value of 2.5 to compare the squared normal value associated with the “best” set to the squared normal value of the “second best” set. After ensuring that the ratio test has passed, the correct integer ambiguity set is passed to the high-rate output task.

The high-rate output task receives data from multiple sources. First, the raw GPS measurements are received from the front-end parser at a 20 Hertz rate. Second, the unit line-of-sight vector from the mobile UAV to each of the SVs in view is received from the Kalman filter. Finally, after the ratio test has passed, the integer value of the widelane ambiguities is received from the ambiguity resolution routine. After the widelane integer ambiguities have been received, the high-rate output task is capable of internally determining the integer ambiguities for any new SVs. The notable exception is the case when an excessive number of SVs are simultaneously lost. However, in this case, the Kalman filter retains the capability of passing the floating point estimates, and covariance matrix, to the ambiguity resolution routine. Thus, the entire integer ambiguity set can be determined from scratch whenever

necessary. Finally, for added precision, after the widelane ambiguities have been determined, a relative position is formulated which permits the computation of the L1 only ambiguity set. This L1 only ambiguity is then used to formulate a more accurate relative position which is then used to determine the narrowlane ambiguity set. Finally, the narrowlane ambiguity set is used to determine the most precise relative position. Of note, the process of stepping from widelane to L1 only, and finally to narrowlane measurements only has to be completed once. The high-rate output task uses the known narrowlane ambiguities to compute the ambiguities for any newly acquired SVs.

## **5.2 *Conclusions***

Because the Kalman filter provides floating point estimates of the double-difference carrier phase ambiguities, and a covariance matrix, to the ambiguity resolution routine, an analysis of the tuning parameters was performed. The state vector errors generally fell within the expected range represented by the covariance matrix.

A properly functioning Kalman filter should significantly reduce the noisy nature of the GPS measurements. To evaluate performance in this area, two relative position solutions were formed. The first solution was obtained by merely taking the difference between the two absolute positions computed by the NovAtel GPS receiver. The second solution was obtained directly from the Kalman filter state estimate of relative position. While the first solution was quite noisy, having significant fluctuations over a short time span, the Kalman filter estimated relative position was much less oscillatory in nature.

A final test of the Kalman filter involved analyzing the performance of the relative velocity and relative acceleration in a dynamic environment. While utilizing a modified golf cart, a series of turns was made while simultaneously logging the state vector data. Post-processing analysis revealed that the estimated velocities and estimated position computed by the Kalman filter were consistent with each other, as well as with the path over the ground.

Because the floating point estimates of the double-difference carrier-phase ambiguities must be resolved to their integer values to obtain the most precise relative position, performance of the ambiguity resolution routine was conducted. The two main performance parameters were the time to resolve the ambiguities and the success rate of actually choosing the correct set.

In terms of the time required to resolve the ambiguities, a test was conducted which involved a sample size of 6,876 calls to the ambiguity resolution routine in a wide variety of measurement conditions. The ambiguities were resolved within 2 seconds for all but 126 cases (98 percent). Additionally, the ambiguities were resolved in 5 seconds or less for all but 13 cases (99.8 percent). Finally, the longest time to resolve the ambiguities was just under two minutes (117 seconds); however, most of the longer convergence times were much less than 117 seconds.

In terms of success rate, of the 6,876 calls to the ambiguity resolution routine, there were only 7 instances where the incorrect ambiguity was returned. This results in a 99.9 percent success rate. Achieving a 100 percent success rate is possible by choosing a higher critical value. However, doing so results in much longer convergence times. Additionally, even when the incorrect set was returned, the error in the relative position was on the order of a meter. Given the low probability of selecting the incorrect set, the critical value was left at 2.5 which results in acceptable accuracy and very fast convergence times.

As will be discussed below, the precision obtained through the use of widelane measurements was less than adequate with errors frequently in the 10-15 cm range. Performance can be significantly improved by using either L1 only or narrowlane measurements. However, it was determined that the time required to resolve the ambiguities was significantly higher when ambiguity resolution was attempted with measurements other than widelane. Additionally, the success rate was moderately lower as well. For this reason, ambiguity resolution is always accomplished with

widelane measurements. To achieve the desired level of precision, the high-rate output task transitions from widelane to L1 only, and finally to narrowlane measurements.

There are two key performance parameters for the high-rate output task. First, the relative position must be accurate to within a few centimeters. Second, the latency of the computed solution must be kept as small as possible.

In terms of precision, the use of widelane resulted in a horizontal accuracy of roughly 8-10 cm and a vertical accuracy of roughly 15-20 cm. An in-depth analysis resulted in the determination that these errors were caused by multipath. As a result, the high-rate output task utilizes the relative position obtained from using the wide-lane measurements to formulate the ambiguity set for L1 only measurements. This ambiguity set is then used to obtain a more accurate relative position which is used to formulate the ambiguity set for narrowlane measurements. It was determined that making the transition from widelane directly to narrowlane was not possible, because the relative position obtained with widelane was not accurate enough to consistently obtain the correct narrowlane ambiguity set. However, the transition to narrowlane mode only has to be accomplished once. After the narrowlane ambiguity set has been determined, the system remains in narrowlane, unless a reset is required. Should a new SV come into view, the high-rate output task utilizes the known ambiguities to calculate a precise relative position. This relative position is then used to formulate the integer ambiguity for the new SV which is then saved for future epochs.

In terms of latency, initial testing resulted in a latency of approximately 10 ms. This is well below the required levels for 20 Hertz output. However, it indicates that the system won't operate reliably if 100 Hertz output is desired. However, additional testing indicated that the 10 ms latency was completely due to the added processing that occurs at a 1 Hertz interval when the Kalman filter is processing data. Neglecting the overhead of the Kalman filter, the system latency is reduced to a range of 2-4 ms. Several recommendations are made in the next section which should permit the system to successfully operate at a significantly faster rate of 100 Hertz.

Time constraints did not allow for dynamic testing as extensive as was done for static testing. However, the limited testing which was conducted revealed that the system was both reliable and accurate in the real-time dynamic environment. Recommendations for follow-on testing are provided in the next section.

One of the initial concerns at the beginning of the research effort was the implication of utilizing data with a 1 Hertz update rate in a 20 Hertz process. Specifically, the unit line-of-sight vector from the wing UAV (mobile receiver) to each of the SVs in view is only calculated at a 1 Hertz rate. This information is required in order to formulate the measurement matrix. The reason for calculating this vector at a reduced rate is because of the added latency inherent in the Kalman filter which was discussed above. However, an analysis of the precision of the high-rate output revealed that the one second latency induced into the measurement matrix was not even noticeable.

### ***5.3 Recommendations***

Unfortunately, time constraints did not permit the system to be ported over to a PC-104 or GUMSTIX hardware configuration. There are no known issues that would preclude the system from operating on such a system. In particular, the GUMSTIX systems have the LINUX operating system permanently burned on the ROM and also support multithreading. However, the testing discussed in Chapter 4 should be repeated on the applicable hardware to ensure that the system continues to operate in an acceptable manner.

A major bottleneck which was identified during development and testing was the transmission of the data from the lead aircraft to the wing aircraft. The current implementation utilizes the NovAtel RANGECMPB data log. This means that for each SV in view, the system presented in this thesis must transmit the following data: L1 lock time, L2 lock time, L1 pseudorange measurement, L2 pseudorange measurement, L1 carrier-phase measurement, L2 carrier-phase measurement, L1 doppler frequency, L2 doppler frequency, and other miscellaneous information contained in the NovAtel log. For a baseline of 12 SVs this required nearly 100 Kbps with the system

operating at 20 Hertz. Several possibilities exist to increase throughput. The most obvious choice is to replace the serial data link system with a faster USB or Ethernet system that has higher data transmission capabilities. However, there are some other alternatives which could prove to be viable. Because the NovAtel RANGECPB data log contains a great deal of data, a pre-processor could be placed onboard the lead aircraft that extracts the information from the NovAtel log and only transmits the necessary data. For example, the L1 and L2 lock times which require 42 bits per epoch for each SV (10 Kbps for 12 SVs at 20 Hertz) could be replaced by a single bit per epoch for each SV (0.24 Kbps for 12 SVs at a 20 Hertz rate) and there would be no impact on cycle slip detection. The L1 doppler frequency and L2 doppler frequency measurements could be completely eliminated yielding a savings of 56 bits per SV per epoch (a savings of 13 Kbps for 12 SVs at 20 Hertz). Even though the Kalman filter operates at a 1 Hertz rate and is the only task that requires code measurements, both L1 and L2 code measurements are transmitted at a 20 Hertz rate. By only transmitting the L1 code measurements, at a reduced 1 Hertz rate, a savings of 16.8 Kbps can be realized for 12 SVs at 20 Hertz.

In terms of latency, it is desired to reduce the overall system from 10 ms to roughly 5-6 ms. This would allow the system to process data at a 100 Hertz rate. The limiting factor for system latency is the Kalman filter which has to calculate the inverse of relatively large matrices (22x22). The instrumentation package currently gives the Kalman filter thread the same priority as the high-rate output thread. However, it should be relatively easy to lower the thread priority associated with the Kalman filter since it has 1 second to complete its calculations before it receives the next set of measurements. Unfortunately time did not permit the completion of this task.

As mentioned in Chapter 4, there was insufficient time to conduct adequate testing of the transition from widelane to L1 only and from L1 only to narrowlane measurements. Preliminary indications revealed that the accuracy obtained in when utilizing widelane measurements was accurate enough to successfully determine the L1 only ambiguity set. Additionally, the accuracy of the L1 only solution appeared

to be accurate enough to correctly determine the narrowlane ambiguity set. It also appeared that going directly from widelane to narrowlane was not possible because the accuracy of the widelane solution was not accurate enough to consistently select the proper narrowlane ambiguity set. However, more testing needs to be undertaken regarding the success rates of transitioning from widelane to narrowlane.

Because cycle slips occur very infrequently during static testing combined with the fact that dynamic testing was relatively limited due to time constraints, more testing is needed to evaluate cycle slip detection and handling. It was assumed that the NovAtel signal lock times would reliably indicate a cycle slip. If this is a valid assumption, then cycle slips can artificially be created by modifying the signal lock times in the file utilized during post-processed mode. However, testing needs to be conducted utilizing actual cycle slips to ensure that the signal lock time provides a reliable indication of cycle slips.



## Appendix A. Software Documentation

This appendix provides a top-level description of how the software is organized. It is hoped that the material to follow will be beneficial to those desiring to utilize and/or enhance the instrumentation package presented in this thesis.

### A.1 File Listing

All source code was written in C++ in a SUSE Linux environment. Table A.1 provides a listing of all required files.

Table A.1: Software Development: File Listing

File Name	Brief Description
amb_res.cpp	Contains functions for ambiguity resolution task
amb_res.h	Header file for amb_res.cpp
cserial.cpp	Contains functions relating to serial ports I/O
cserial.h	Header file for cserial.cpp
data.cpp	Contains class definitions for various data types
data.h	Header file for data.cpp
filter.cpp	Contains functions for Kalman filter task
filter.h	Header file for filter.cpp
high_rate.cpp	Contains functions for high-rate output task
high_rate.h	Header file for high_rate.cpp
lambda.cpp	Contains functions for C++ implementation of LAMBDA routine
lambda.h	Header file for lambda.cpp
parse.cpp	Contains functions for front end parser
parse.h	Header file for parse.cpp
project.cpp	Contains function for top level program
project.h	Header file for project.cpp

Additionally, an open source matrix library was obtained and utilized. The matrix library, designed by Robert Davies, can be downloaded from his website [3] and contains extensive documentation on the various matrix functions which were used.

## ***A.2 Software Organization***

The program initializes by calling the “main” function which is located at the end of the “project.cpp” source code file. Depending on the operating mode selected by the user, one of two paths are taken.

In post-processed mode, the “main” function generates two threads which operate simultaneously. The first thread, “filter\_thread”, is for the Kalman filter task and is located in the “filter.cpp” source code file. The second thread, “high\_rate\_thread”, is for the high-rate output task and is located in the “high\_rate.cpp” source code file. Once the two previously mentioned threads have been initialized, the “main” function passes control to the “process\_data\_file” function which is located in the “parse.cpp” source code file. This function reads the raw measurements from the “common.txt” input file and passes the data to the appropriate thread in the same manner as the “Parse\_thread” does when operating in the real-time mode discussed next.

In real-time mode, the “main” function generates three threads which operate simultaneously. The first thread, “COM1\_thread”, is located in the “project.cpp” source code file and handles communication with the reference receiver (lead aircraft via serial data link). The second thread, “COM2\_thread”, is also located in the “project.cpp” source code file and handles communication with the mobile receiver (wing aircraft via serial cable). The third thread, “Parse\_thread” is located in the “parse.cpp” source code file and is used to synchronize the raw GPS measurements. The “Parse\_thread” generates the “common.txt” input file which is required in order to be able to operate in the previously discussed post-processed mode. Finally, the “Parse\_thread” generates two additional sub-threads. These are the “filter\_thread” and the “high\_rate\_thread” and are identical to those discussed in the post-processed mode section.

Both the “COM1\_thread” and the “COM2\_thread” are nearly identical in nature. The only difference between the two is that the first handles communication with the lead aircraft while the second handles communication with the wing air-

craft. After initializing the appropriate communication port, each thread requests the RANGECMPB, GPSEPHEMB, and BESTPOSB NovAtel data logs be sent at the required intervals. After a data log is sent by the NovAtel, each thread calls the “process\_HEADER” function, located in the “project.cpp” source code file, to decode the 28 byte header. Depending on the message type contained in the header, one of the following functions are called to decode the remainder of the data log: “Process\_RANGECMPB”, “Process\_GPSEPHEMB”, or “Process\_BESTPOSB”. Each of these files is located in the “project.cpp” source code file. Once a complete data log has been received and decoded, the data is pushed onto a memory stack. After the “Parse\_thread” synchronizes the measurements, the data is popped off the stack and passed to the “filter\_thread” at a 1 Hertz rate and to the “high\_rate\_thread” at a 20 Hertz rate.

The “filter\_thread”, located in the filter.cpp source code file, contains several subroutines that are called at various time intervals while the Kalman filter is operating. These are summarized in Table A.2 and can be found in the “filter.cpp” source code file.

Table A.2: Kalman Filter Thread: List of Subroutines

Function Name	Brief Description
lla2ecef	Converts longitude, latitude, and altitude into ECEF coordinates
get_filter_params	Reads the user defined Kalman filter tuning parameters from the “input.txt” file
init_matrix_1	Initializes $\Phi$ , $Q_d$ , and $R$ matrices
init_matrix_2	Initializes $P$ matrix
calc_sv_pos	Calculates SV position in ECEF coordinates. Required to compute $H$ matrix.
Generate_T_Matrix	Generates $T$ matrix. Used to transform $x$ and $P$ matrices when base SV is lost

The “filter\_thread” also calls the “amb\_resolution” function which is located in the “amb\_res.cpp” source code file. In turn, the “amb\_resolution” function calls the “lambda2” function. The “lambda2” function and all of its subroutines are contained in the “lambda.cpp” source code file.

The “high\_rate\_thread”, located in the high\_rate.cpp source code file, contains the code which calculates the relative position. It receives data from three sources. First, the “filter\_thread” provides the unit line-of-sight vector to each SV in view. Second, the “amb\_resolution” function passes the integer ambiguity values after the ratio test has been passed. Finally, the “Parse\_thread” passes the raw GPS measurements at a 20 Hertz rate. It should be noted that after the fixed ambiguities are initially received, any new ambiguities (from newly acquired SVs) are internally determined from within the “high\_rate\_thread”.

### ***A.3 Implementation Aspects***

In order to successfully compile the program, there are two compiler flags that must be set within the KDE environment. First, because the application utilizes multi-threading, the -pthread flags must be set. Second, because the application utilizes the previously discussed matrix library, the path containing the matrix library must also be included in the compiler flags directives.

## Bibliography

1. ANTCOM Inc. website. "Active/Passive, L1/L2 GPS Antennas", <http://www.antcom.com/PDF-8-1999/PN9D.pdf> (11 April 2006).
2. Bouska, T.J. *Development and Simulation of a Pseudolite-Based Flight Reference System*. Masters thesis, Air Force Institute of Technology, 2003.
3. Davies, R. "Newmat C++ matrix library", <http://www.robertnz.net/nm.intro.htm> (27 August 2006).
4. FreeWave Inc. website. "Radio Products", <http://www.antcom.com/PDF-8-1999/PN9D.pdf> (11 April 2006).
5. Jonge de, P.J., Tiberius, C.C.J.M. *The LAMBDA Method for Integer Ambiguity Estimation: Implementation Aspects*. Technical report, Delft University of Technology, Geodetic Computing Centre LGR series, 1996.
6. Joosten, Peter. *The LAMBDA Method - MATLAB Implementation*. Delft University of Technology, Netherlands, March 2001.
7. Lachapelle, G., Cannon, M.E. Lu, G., Sun, H. *Precise Aircraft-to-Aircraft Positioning Using a Multiple Receiver Configuration*. National technical meeting, Institution of Navigation, San Diego CA, January 1994.
8. Maybeck, P. *Stochastic Models, Estimation, and Control*. Mathematics in Science and Engineering, Arlington VA, 1994.
9. Misra, Enge P., P. *Global Positioning System: signals, measurements, and performance*. Ganga-Jamuna Press, Lincoln MA, 2001. ISBN 0970954409.
10. NavtechGPS Inc. website. "40dB L1/L2 Antenna by Antcom", <http://www.navtechgps.com/supply/L1L2roof.asp> (11 April 2006).
11. NovAtel Inc. *OEM4 Family, User Manual - Volume I, Installation and Operation*, revision 17 edition, August 2005.
12. Online Positioning User Service (OPUS) website, <http://www.ngs.noaa.gov/OPUS/> (17 August 2006).
13. Parkinson, Spilker Jr. J.J., B.W. *Global Positioning System: Theory and Applications*, volume 1. American Institute of Aeronautics and Astronautics, Washington DC, 1996. ISBN 156347106X.
14. Parkinson, Spilker Jr. J.J., B.W. *Global Positioning System: Theory and Applications*, volume 2. American Institute of Aeronautics and Astronautics, Washington DC, 1996. ISBN 156347106X.
15. Raquet, John F. *Development of a Method for Kinematic GPS Carrier-Phase Ambiguity Resolution Using Multiple Reference Receivers*. Phd dissertation, University of Calgary, 1998.

16. Raquet, John F. Class handouts distributed in EENG-533, EENG-533: Navigation Using the Global Positioning System, April 2005.
17. Raquet, John F. Class handouts distributed in EENG-633, EENG-633: Advanced GPS Theory and Application, June 2005.
18. Spinelli, C.J. *RTK Precise DGPS Positioning Algorithm Between Two Aircraft*. Masters thesis, Air Force Institute of Technology, 2006.
19. Teunissen, P.J.G. “The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer estimation”. *Journal of Geodesy*, 70(1-2):65–82, November 1995.
20. Teunissen, P., Verhagen, S. *On the Foundation of the Popular Ratio Test for GNSS Ambiguity Resolution*. Technical report, Institute of Navigation, GNSS, 17th International Technical Meeting of the Satellite Division, September 2004.
21. Welch, Bishop G., G. *An Introduction to the Kalman Filter*. Technical Report TR 95-041, University of North Carolina, Chapel Hill NC, April 2004.
22. Williamson, W.R. *Real Time, High Accuracy, Relative State Estimation for Multiple Vehicle Systems*. Ph.D. thesis, University of California, Los Angeles, 2000.
23. Zarchan, Musoff H., P. *Fundamentals of Kalman Filtering: A Practical Approach*, volume 208. American Institute of Aeronautics and Astronautics, Inc., second edition, 2005.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YYYY) 14-09-2006		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) Aug 2004 - Sep 2006	
4. TITLE AND SUBTITLE Development of a Low-Latency, High Data Rate, Differential GPS Relative Positioning System for UAV Formation Flight Control				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Comstock, Stephen J., CDR, USN				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GAE/ENG/ 06-03	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Dr. Mikel M. Miller, AFRL / SNRN Bldg. 620, Room 3AJ39 2241 Avionics Circle WPAFB, OH 45433-7333 (937)-255-6127 x-4274 Mikel.Miller@wpafb.af.mil				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>In order for Unmanned Aerial Vehicles (UAVs) to be able to fly missions currently performed by manned aircraft, they must be able to conduct in-flight refueling. Additionally, significant fuel savings can be realized if multiple UAV's are able to fly in precise formation and align wingtip vortices. In either case, the precise relative position between the aircraft must be known to an accuracy of only a few centimeters.</p> <p>Previous research at the Air Force Institute of Technology culminated in the development of a relative positioning system for manned aircraft. This thesis presents the development of the next-generation system designed for small UAV's. Because of the stringent size, weight, and power consumption requirements inherent in small UAV's, several approaches were taken to maximize efficiency and performance while simultaneously keeping the system small and lightweight.</p> <p>At the core of the Differential GPS (DGPS) application presented in this thesis are three separate tasks which operate asynchronously yet share information when required. A Kalman filter task operates continuously at a 1 Hertz rate. An ambiguity resolution task, utilizing the Least squares AMBIGUITY Decorrelation Adjustment (LAMBDA) method, is run whenever the floating point ambiguities must be resolved to their integer values. A high-rate output task, operating at a 20 Hertz rate, formulates a high-rate, centimeter-level, relative position solution with less than 10 milliseconds of latency. The use of widelane measurements generally resulted in a 2 second convergence time for ambiguity resolution and a 99.9 percent success rate of selecting the proper ambiguity set. However, in order to minimize the increased errors associated with multipath, the system quickly transitions from widelane mode to narrowlane mode. The system was tested on the ground in both a static and dynamic environment. Unfortunately there was inadequate time to conduct flight testing using radio controlled aircraft to simulate small UAV's.</p>					
15. SUBJECT TERMS GPS, Differential GPS (DGPS), Kalman Filtering, Ambiguity Resolution, Least Squares AMBIGUITY Decorrelation Adjustment (LAMBDA) Method, Unmanned Aerial Vehicle (UAV), Formation Flight Control, Relative Positioning.					
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT			c. THIS PAGE	19b. TELEPHONE NUMBER (Include area code)
U	U	UU	130	Dr. John F. Raquet, AFIT / ENG (937) 255-6565, ext 4580; e-mail: John.Raquet@afit.edu	